Università di Genova

Dipartimento di
Informatica, Bioingegneria,
Robotica e Ingegneria dei Sistemi

Graphical Models for Multivariate Time-Series

Federico Tomasi

Theses Series DIBRIS-TH-2019-4

Università di **Genova**

Dipartimento di Informatica, Bioingegneria,
Robotica ed Ingegneria dei Sistemi

Ph.D. Thesis in
Computer Science and Systems Engineering
Computer Science Curriculum

Graphical Models for Multivariate Time-Series

by

Federico Tomasi

March, 2019

Ph.D. Thesis in Computer Science and Systems Engineering (S.S.D. INF/01)
Dipartimento di Informatica, Bioingegneria, Robotica ed Ingegneria dei Sistemi
Università di Genova

Candidate
Federico Tomasi
federico.tomasi@dibris.unige.it

Title
Graphical Models for Multivariate Time-Series

Advisors
Annalisa Barla
DIBRIS, Università di Genova
annalisa.barla@unige.it

Alessandro Verri
DIBRIS, Università di Genova
alessandro.verri@unige.it

External Reviewers
Mauricio A. Álvarez Lopez
Department of Computer Science, University of Sheffield
mauricio.alvarez@sheffield.ac.uk

Neil Lawrence
Department of Computer Science, University of Sheffield and The Sheffield Institute for Translational Neuroscience
n.lawrence@sheffield.ac.uk

Location
DIBRIS, Univ. di Genova
Via Opera Pia, 13
I-16145 Genova, Italy

Submitted On
March 2019

If Nature abhors the void,
the mind abhors what is meaningless.

— Arthur Koestler, *The Ghost in the Machine*

To my family

Abstract

Gaussian graphical models have received much attention in the last years, due to their flexibility and expression power. In particular, lots of interests have been devoted to graphical models for temporal data, or dynamical graphical models, to understand the relation of variables evolving in time. While powerful in modelling complex systems, such models suffer from computational issues both in terms of convergence rates and memory requirements, and may fail to detect temporal patterns in case the information on the system is partial. This thesis comprises two main contributions in the context of dynamical graphical models, tackling these two aspects: the need of reliable and fast optimisation methods and an increasing modelling power, which are able to retrieve the model in practical applications. The first contribution consists in a forward-backward splitting (FBS) procedure for Gaussian graphical modelling of multivariate time-series which relies on recent theoretical studies ensuring global convergence under mild assumptions. Indeed, such FBS-based implementation achieves, with fast convergence rates, optimal results with respect to ground truth and standard methods for dynamical network inference. The second main contribution focuses on the problem of latent factors, that influence the system while hidden or unobservable. This thesis proposes the novel *latent variable time-varying graphical lasso* method, which is able to take into account both temporal dynamics in the data and latent factors influencing the system. This is fundamental for the practical use of graphical models, where the information on the data is partial. Indeed, extensive validation of the method on both synthetic and real applications shows the effectiveness of considering latent factors to deal with incomplete information.

Publications

Some ideas and figures have appeared previously in the following publications, which cover most of the work developed during my Ph.D. studies.

Articles

Squillario, Margherita, Federico Tomasi, Veronica Tozzo, Annalisa Barla, and Daniela Uberti (2018). "A 3-fold kernel approach for characterizing Late Onset Alzheimer's Disease". In: *bioRxiv*, p. 397760.

Tozzo, Veronica, Federico Tomasi, Margherita Squillario, and Annalisa Barla (Nov. 2018). "Group induced graphical lasso allows for discovery of molecular pathways-pathways interactions". In: *Machine Learning for Health (ML4H) Workshop at NeurIPS 2018*. arXiv: 1811.09673 [q-bio.QM].

Conference Proceedings

Barbieri, Matteo, Samuele Fiorini, Federico Tomasi, and Annalisa Barla (2016). "PALLADIO: A Parallel Framework for Robust Variable Selection in High-Dimensional Data". In: *6th Workshop on Python for High-Performance and Scientific Computing, PyHPC@SC 2016, Salt Lake, UT, USA, November 14, 2016*. IEEE, pp. 19–26. DOI: 10.1109/PyHPC.2016.007. URL: https://doi.org/10.1109/PyHPC.2016.007.

D'Amario, Vanessa, Federico Tomasi, Veronica Tozzo, Gabriele Arnulfo, Annalisa Barla, and Lino Nobili (2018). "Multi-task multiple kernel learning reveals relevant frequency bands for critical areas localization in focal epilepsy". In: *Proceedings of the 3rd Machine Learning for Healthcare Conference*. Ed. by Finale Doshi-Velez, Jim Fackler, Ken Jung, David Kale, Rajesh Ranganath, Byron Wallace, and Jenna Wiens. Vol. 85. Proceedings of Machine Learning Research. Palo Alto, California: PMLR, pp. 348–382. URL: http://proceedings.mlr.press/v85/d-amario18a.html.

Fiorini, Samuele, Federico Tomasi, Margherita Squillario, and Annalisa Barla (2019). "Adenine: A HPC-Oriented Tool for Biological Data Exploration". In: *Computational Intelligence Methods for Bioinformatics and Biostatistics*. Ed. by Massimo Bartoletti, Annalisa Barla, Andrea Bracciali, Gunnar W. Klau, Leif Peterson, Alberto Policriti, and Roberto Tagliaferri. Cham: Springer International Publishing, pp. 51–59. ISBN: 978-3-030-14160-8.

Tomasi, Federico, Veronica Tozzo, Alessandro Verri, and Saverio Salzo (2018a). "Forward-Backward Splitting for Time-Varying Graphical Models". In: *Proceedings of the Ninth International Conference on Probabilistic Graphical Models*. Ed. by Václav Kratochvíl and Milan Studený. Vol. 72. Proceedings of

Machine Learning Research. Prague, Czech Republic: PMLR, pp. 475–486. URL: http://proceedings.mlr.press/v72/tomasi18a.html.

Tomasi, Federico, Veronica Tozzo, Saverio Salzo, and Alessandro Verri (2018b). "Latent Variable Time-varying Network Inference". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '18. London, United Kingdom: ACM. ISBN: 978-1-4503-5552-0. DOI: 10.1145/3219819.3220121. URL: http://doi.acm.org/10.1145/3219819.3220121.

Tomasi, Federico, Margherita Squillario, Alessandro Verri, Davide Bagnara, and Annalisa Barla (2019). "ICING: large scale inference of immunoglobulin clonotypes". In: *Computational Intelligence Methods for Bioinformatics and Biostatistics*. Ed. by Massimo Bartoletti, Annalisa Barla, Andrea Bracciali, Gunnar W. Klau, Leif Peterson, Alberto Policriti, and Roberto Tagliaferri. Cham: Springer International Publishing, pp. 42–50. ISBN: 978-3-030-14160-8.

Oral Communication and Posters

Squillario, Margherita, Matteo Barbieri, Samuele Fiorini, Federico Tomasi, and Annalisa Barla (Apr. 2017). *Uncovering Alzheimer's SNP signature with a multi-view machine learning analysis based on SNPs, genes and pathways*. Alzheimer's & Parkinson's Diseases Congress - AD/PD Vienna 2017.

Tomasi, Federico, Margherita Squillario, and Annalisa Barla (Aug. 2017). *Knowledge Driven Graph Evolution (KDGE)*. F1000Research 2017, 6(ISCB Comm J):1380 (poster). DOI: 10.7490/f1000research.1114633.1. Presented at Joint 25th Annual International Conference on Intelligent Systems for Molecular Biology (ISMB) and 16th European Conference on Computational Biology (ECCB) 2017.

Acknowledgements

I would gratefully thank all of the people that helped during the development of the work presented in this thesis.

Firstly, I would like to thank my advisors Annalisa Barla and Alessandro Verri and my research group at Università di Genova. Special thanks go to Veronica Tozzo for contributing to Section 4.1 and Chapter 5, Margherita Squillario, for biological considerations of Section 6.2.1, Vanessa D'Amario, for the analysis of SEEG data in Section 8.2.
I am deeply grateful to Saverio Salzo for the great help in developing the main contributions of this thesis of both Chapters 4 and 5.
A mention to the research group based in Manchester, headed by Magnus Rattray, in particular to Mudassar Iqbal for discussions on graphical models applied to biological data, and to Muhammad Fadlullah Wilmot, for some of the figures and biological considerations of Chapter 7. Thanks to Mauricio Alvarez, for the original Wishart Process code (Chapter 8).

And to all the people that indirectly contributed to this thesis.

To the colleagues of R309, in particular Angelo, Laura and Tommaso, for sharing this venture since the very start.

To Ginevra, for a constant support, endless patience and profound love.

And to my family, for their continuous encouragement that made all this possible.

Contents

List of Figures

List of Tables

List of Algorithms

Introduction

Recent developments in data storage and computing techniques led to a massive amount of measurements in a wide set of applicative areas, such as finance, sociology and genomics. A problem which arises in data analysis is that most of the variables which describe one sample may interact in different ways, exhibiting a wide range of peculiar variability patterns across samples. Throughout this thesis, I will refer to such kind of data, which comprise samples described by an high number of variables, as *high-dimensional*.

High-dimensional data are difficult to describe in a parsimonious model. Indeed, searching for complex patterns in the data may offer insights on the behaviour of variables in diverse contexts, such as different biological conditions in biomedical studies. Often, the interest is to understand interaction patterns of such variables included in a system. Interactions are usually modelled as a network (or graph), *i.e.*, a set of variables (nodes) connected with each other based on a particular type of relation (links). The graphical modelling of the variables offers a compact and efficient representation which helps to identify the variability patterns in the data.

However, dealing with high-dimensional data implies several drawbacks. A fundamental limitation is that high-dimensional data require a large amount of samples to reliably capture the variance of the variables. Indeed, especially in (but not limited to) biological systems, variables may be hundreds of thousands, while samples describing them are just a few (this is usually referred to as a $d \gg n$ scenario, with d dimensions and n samples). An efficient strategy relies on restricting the complexity of the resulting model, which improves the robustness to noise while increasing the interpretability of the results, providing insights on the underlying processes of the system. This notion is often referred to as *sparsity*, a main concept throughout this thesis introduced in details in Section 1.2.

Also, when the dimensionality d increases, the intuition of locality does not hold any more. Hence, machine learning methods for pattern discovery that rely on Euclidean distances are not directly applicable. This is a known problem commonly referred to as the *curse of dimensionality* (Friedman, Hastie, and Tibshirani, 2001).

Indeed, starting from a set of high-dimensional data, the goal of pattern discovery is to find a model which is complex enough to capture data variability while still being interpretable, which helps the model validation where the ground truth is not known, and robust to noise, in contrast to the restricted number of samples available in practical contexts, possibly exploiting prior knowledge on the underlying processes to efficiently direct the inference of the model.

Motivation

This thesis focuses on the structure learning problem of high-dimensional and temporal data, under the presence of hidden conditioning factors. During the last years the problem of uncovering an underlying structure, *e.g.*, an interaction graph between variables, has received much attention, particularly for the availability of an always increasing number of samples.

Common approaches rely on pairwise similarity measures between variables, such as mutual information scores. This leads to pairwise network inference methods (Chapter 1) that rely on local similarities, hence lacking a solid way to assess the global inferred network. For example, a leading strategy is to limit the number of links between variables based on an arbitrary threshold, to avoid an over-representation of the network.

Instead, probabilistic graphical models represent a theoretically grounded framework for network inference, which aim to describe high-dimensional data under a parsimonious model. In this case the underlying graph describes the conditional independence among a set of random variables, which are assumed to follow a joint probability distribution. Such graphical representation has numerous advantages in lots of machine learning areas. For example, the graph indicates the joint relevance of groups of features, which can be exploited to improve a predictive model (Hernández-Lobato, Hernández-Lobato, and Suárez, 2011).

A lot of interest, recently, has been also devoted to longitudinal data, for example in the context of neuronal activity or volatility analysis. Such applications involve a developing system over time, which can be visualised by an ever changing structure of the underlying network between the variables that describes the system. The modelling of multivariate time-series as the evolution of a dynamical system may be beneficial to understand underlying processes which generate the system.

A further challenge in data analysis is that data are usually subject to latent (*i.e.*, hidden or unmeasurable) factors which influence the majority of the system while not being part of the system itself. This may be caused by missing or incomplete information on the data under analysis, a relevant assumption for the analysis of real systems.

This thesis tackles the graphical modelling problem of multivariate time-series, focusing on three main aspects: *(i)* the evolving dynamics of multivariate time-series and their relations, *(ii)* the presence of global hidden factors, and *(iii)* the use of appropriate optimisation methods able to efficiently deal with the inference of complex models.

Contribution

This thesis revolves around two main contributions, related to the context of graphical modelling of time-series data. The first core contribution consists in two efficient algorithms based on the forward-backward splitting for

the time-varying graphical lasso model, a method for the inference of multiple networks in time. This is motivated by the need of efficient algorithms that can deal with complex models, while at the same time being able to describe high-dimensional data. Such algorithms rely on recent advances of the forward-backward splitting, representing the starting point of this contribution, which outperforms, for common use cases, standard optimisation methods for graphical inference under complex temporal models.

The second core contribution consists in the latent variable time-varying graphical lasso, a novel method for the network inference of time-series data subject to latent (*i.e.*, hidden or unmeasurable) factors. Such machine learning method for multivariate time-series data answers to the problem of describing an evolving system subject to the presence of global hidden factors, which influence the system without being explicitly measured.

All contributions are extensively validated on synthetic data, testing their efficiency and performance. Also, numerous applications widely illustrate the relevance of the latent variable time-varying graphical lasso on real data, to show the practical use and advantage of considering latent factors during the inference of a dynamical model with both biological and financial data.

Notation

Unless otherwise specified, real-valued variables are denoted with lower case letters (such as x). Uni-dimensional vectors are denoted by boldface lower case letters (such as $\boldsymbol{x}$). Matrices (*i.e.*, vectors with 2 dimensions) are denoted by non-bold upper case letters (such as X). Tensors (*i.e.*, vectors with more than 2 dimensions) are denoted by boldface upper case letters (such as $\boldsymbol{X}$). Vectors, matrix or tensor entries are denoted by subscript (such as $x_{i,j}$ or x_{ij} — whether clear from the context, the comma can be omitted).

A special notation may be used to highlight the single variables. An element of $\mathbb{R}^{|\Gamma|}$ is denoted by

$$\boldsymbol{x} = \boldsymbol{x}_\Gamma = (x_1, \dots, x_{|\Gamma|}) = (x_i)_{i \in \Gamma}.$$

Similarly, an element of $\mathbb{R}^{|\Gamma| \times |\Gamma|}$ (a square matrix) is denoted by

$$X = X_\Gamma = (x_{ij})_{i,j \in \Gamma}.$$

The trace of a matrix is indicated by $\operatorname{tr}(X)$. $\mathcal{S}^d$ indicates the cone of symmetric $d \times d$ matrices, so that $\mathcal{S}^d \subset \mathbb{R}^{d \times d}$. $\mathcal{S}^d_+$ denotes the cone of symmetric $d \times d$ positive semi-definite matrices, and similarly $\mathcal{S}^d_{++}$ the cone of square symmetric $d \times d$ positive-definite matrices. Furthermore, for every square symmetric matrix $X \in \mathcal{S}^d$, $X \succ 0$ means that X is positive definite (or, equivalently, $X \in \mathcal{S}^d_{++}$), and $X \succeq 0$ means that X is positive semi-definite (or, equivalently, $X \in \mathcal{S}^d_+$).

$\mathcal{H}$ denotes a generic Euclidean space, and by $\langle \cdot, \cdot \rangle$ its scalar product. $\|\cdot\|$ is the standard ℓ_2-norm. When the argument of the norm is a matrix (or a tensor), *i.e.*, $\|\boldsymbol{X}\|$, the norm is the Frobenius norm (often indicated with $\|\cdot\|_F$).

Outline

This thesis comprises four main parts.

Part I contains the background on network inference (Chapter 1), the state of the art on graphical models (Chapter 2), which serves as the basis of the core of the work developed in this thesis, and methods to infer and select a machine learning model (Chapter 3).

Part II includes the main contributions of this thesis, that are the advances on the graphical models for temporal data, namely the time-varying graphical lasso under forward-backward splitting (Chapter 4) and the latent variable time-varying graphical lasso (Chapter 5).

Part III contains the applications on real data of the graphical models for time-series analysis, in particular considering breast cancer evolution (Chapter 6), haematopoietic stem cells (Chapter 7) and epilepsy (Chapter 8).

Lastly, Part IV includes additional mathematical details on graph theory and linear algebra operations that have been used throughout this thesis (Appendix A), and a side project developed in parallel during the work of this thesis (Appendix B).

Part I

Background

This part contains a wide description on the context in which this thesis is posed. Chapter 1 presents two general approaches to the graphical inference problem, with an overview on popular state-of-the-art methods for network inference. Chapter 2 focuses on methods for graphical modelling, which serves as the basis for the main contributions of this thesis. Chapter 3 introduces widely used methods for model optimisation, selection and validation, exploited throughout this thesis.

1 Network Inference

The problem of network inference arises in lots of applications, where the underlying graph structure of variables is not known. In such cases, the interest is to estimate their relations from samples. This task has drawn a lot of attention for example in finance, for volatility analysis, and in computational biology, where the network inference has a crucial role in understanding how molecular interaction works. Indeed, networks pervade all aspects of human health (Barabasi and Oltvai, 2004). In particular, network analysis plays a central role at the cellular level, since most of the cellular components are connected through complex regulatory (Friedman et al., 2000; Hecker et al., 2009; Lozano et al., 2009), metabolic (Kanehisa, 2001) and protein-protein interaction (PPI) networks (Huang, Liao, and Wu, 2016; Jansen et al., 2003).

High-throughput technologies allow to describe samples with a large set of measured variables, that in this context correspond to nodes in a network. Links between variables, instead, correspond to particular relationships which depend on the network considered. Indeed, one can model such links between variables in different ways, corresponding to different network models.

While variables have a different meaning depending on the application, the graph theory at the basis of graphical inference methods remains valid in diverse contexts.

Outline

This chapter introduces two general strategies for network inference, based on pairwise similarity and on probabilistic models. Section 1.1 contains an overview on approaches based on pairwise similarity measures, such as mutual information measures. Section 1.2 sets the network inference problem within a regularised machine learning contexts, which serves as the basis of the graphical models of next chapters. Section 1.3 includes an overview on Bayesian structure learning to infer multiple graphs associated to probability distributions.

1.1 Pairwise Networks

Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of vertices (or nodes) in the network, and $\mathcal{E}$ is the set of edges (or links) between the nodes. In the context of network inference, variables correspond to nodes in the network. Hence, given a set of samples that correspond to realisations of the variables, the goal of pairwise network inference methods is to assign a similarity score between every possible pair of variables across samples. To avoid an over-representation of the network, a possible strategy is then to use a threshold to

retain only the most informative connections in the network. Models which produce such undirected graphs are usually called information theory models (Hecker et al., 2009). Because of their simplicity and low computational cost, they can be used with large-scale networks.

A drawback of such models is that the inference of links between nodes happens iteratively (that is, locally between each pair of nodes), so they do not take into account multiple nodes which can be jointly cause of, for example, a regulation of another node.

A simple network architecture is the so-called *correlation network*, where edges in the network correspond to the weights of correlation coefficients between the nodes (Langfelder and Horvath, 2008; Stuart et al., 2003). Formally, the weight w of the correlation between two nodes (x_i, x_j) in such a network can be generally defined as follows:

$$w_{ij} = \kappa(x_i, x_j), \tag{1.1}$$

for some correlation function κ. Hence, two nodes are said to interact if their correlation is higher than a specific threshold. The threshold allows to increase or decrease the sparsity of the resulting network, to avoid an over-representation of the resulting undirected graph.

Possible extensions of this idea involve different types of scores between the variables. Instead of correlation coefficients, one can use Euclidean distances or information theoretic scores, such as mutual information (Steuer et al., 2002). Mutual information scores are used in lots of popular algorithms for network inference. Examples include:

- RELNET (Butte and Kohane, 2000), which produces relevance networks for functional genomic clustering;
- ARACNE (Margolin et al., 2006), which aims to reconstruct gene regulatory networks, and it has been assessed on human microarray data;
- CLR (Context likelihood of relatedness) (Faith et al., 2007) or its recent extension CLR-MIC (Context likelihood of relatedness with maximal information coefficient) (Akhand et al., 2015).

Also, one can use asymmetric scores to infer directed networks (Rao, Hero, and Engel, 2007).

Correlation networks inference methods have numerous applications in real contexts, due to the ease of implementation and usage. One of the most popular framework for network inference is WGCNA (Langfelder and Horvath, 2008), an R software package which includes a collection of functions for the analysis of weighted correlation networks. In the context of inferring trascriptional networks, other popular algorithms include MINET (Meyer, Lafitte, and Bontempi, 2008), which is implemented in R programming language and uses mutual information to model gene-to-gene interactions. ARACNE-AP (Lachmann et al., 2016) is a new Java implementation of ARACNE (Margolin et al., 2006), which makes use of mutual information estimation between

nodes in order to model a network. ARACNE was extended for time-series data, as in TIMEDELAY-ARACNE (Zoppoli, Morganella, and Ceccarelli, 2010). Other recent software packages include CONET (Faust and Raes, 2016), an extension based on CYTOSCAPE (Shannon et al., 2003), a popular tool for network analysis and visualisation.

1.2 *Sparse Network Inference*

Independently from the measure adopted, pairwise correlation methods infer a single link each time. Also, the resulting network is not sparse *per se*, but low-weighted links are discarded afterwards based on an arbitrary threshold.

A robust approach to infer a network in a global manner is to interpret such task in a machine learning setting. The idea, in practice, is to find the underlying function which generates the data. In a real setting this function is not known, thus can only be approximated by means of some techniques. Hence, this task has given the name of *pattern recognition* (Bishop, 2006).

1.2.1 Regularisation Methods

Regularisation methods are a popular class of techniques to find a function $f_{\boldsymbol{w}}$ which model a set X of observed data, such that, in the case of classification methods, $\hat{Y} = f_{\boldsymbol{w}}(X)$ is as close as possible to the ground truth Y. Such approaches are particularly useful in the $d \gg n$ scenario, where achieving stable solutions is not trivial due to the ill-posedness of the problem (Hastie, Tibshirani, and Friedman, 2009).

For any function $f_{\boldsymbol{w}} : \mathcal{X} \to \mathcal{Y}$ the solution of these methods can be estimated by minimising an objective function in the following form:

$$\underbrace{\ell(\boldsymbol{y}, f_{\boldsymbol{w}}(X))}_{\textit{loss function}} \quad + \quad \lambda \underbrace{\Omega(f_{\boldsymbol{w}})}_{\textit{regularisation penalty}}. \tag{1.2}$$

The loss function estimates the expected risk $\mathcal{E}(f)$:

$$\mathcal{E}(f) = \mathbb{E}[\ell(y, f(x))] = \int p(x, y)\ell(y, f(x))\,\mathrm{d}x\,\mathrm{d}y,$$

which is impossible to compute since the joint probability density function (pdf) $p(x, y)$ is not known. The loss function is a measure of *adherence* to the training data. Instead, the regularisation penalty introduces additional prior information that can be used to solve the problem. The regularisation parameter λ controls the trade-off between the two terms. A common choice for the regularisation penalty is a ℓ_p-norm $\|\boldsymbol{w}\|_p = (\sum_i |w_i|^p)^{1/p}$. Different choices for the value of p produce different effects on the solution (Hastie, Tibshirani, and Wainwright, 2015). In particular, the popular choices $p = 1$ and $p = 2$ lead to the ℓ_1- and ℓ_2-norm, respectively, which will be largely exploited throughout the work of this thesis.

1.2.2 Regularisation Methods for Graphical Inference

Regularisation methods can be applied to graphical inference. In particular, the inference can be stated as a minimisation problem, where the goal is to approximate the true underlying network starting from observations of the system. Clearly, in real cases the true underlying network is not known. Hence, the approach may be to solve a maximum likelihood problem, with the addition of penalty terms to control the complexity of the estimated graph (see Section 2.3). Such methods search for a suitable model describing available samples in a global manner, that is, edges of the graph are estimated at once. An advantage of this approach is the possibility of modelling prior information on the resulting graph. Indeed, such approach is theoretically grounded, relying on the assumption that variables follow appropriate probability distributions. Regularised methods for network inference are shown to be efficient and usable in practice, able to infer networks even in high-dimensional cases.

A desired property of the regularisation penalty, in particular in the context of graphical inference, is that it should enforce *sparsity* in the solution. This idea relates to variable selection in standard regression problems, where the assumption is that the output of interest only depends on a subset of the input variables.

Sparse models are fundamental in lots of applications, in particular where the number of variables is higher than the number of available samples (the so-called $d \gg n$ problem). In this context, while standard statistical guarantees are not available any more, a sparse prior enforces to infer a simpler model, thus helping to its identification, improving the interpretability of the results and reducing the noise. Thanks to their flexibility sparse regularisation methods have been effectively used in biological contexts, dealing with high-throughput data (Giraud, 2014; Mascelli et al., 2013; Mosci et al., 2008; Silver et al., 2013).

Formally, priors on the problem translate into arbitrary choices of the regularisation penalty Ω. In particular, a popular sparsity-enforcing penalty is the so-called ℓ_1-norm. In the context of graphical inference, the ℓ_1-norm penalises the weight of edges between the variables, forcing most of the edges to be zero, thus selecting only a subset of possible connections. The use of ℓ_1-norm is encouraged by the fact that is convex (even though non-smooth).

Indeed, a wide set of methods are based on a lasso-based selection of edges in the graph (Bien and Tibshirani, 2011; Ravikumar et al., 2011; Wainwright, Ravikumar, and Lafferty, 2007; Yuan and Lin, 2007). Among these, notable methods for regularised graphical inference based on the ℓ_1-norm are the neighbourhood-based selection (Meinshausen and Bühlmann, 2006), penalised maximum-likelihood estimation (Banerjee, Ghaoui, and d'Aspremont, 2008) and the graphical lasso (Friedman, Hastie, and Tibshirani, 2008), formalised in details in Chapter 2. The use of a lasso penalty on the edges of the network allows to natively infer sparse graphs, which helps with the interpretability and reduction of noise.

Sparsity-enforcing penalties introduce issues in the practical optimisation of the functionals. To this aim, the main contributions of this thesis rely on

optimisation methods which are natively able to deal with non-smooth penalty terms (Section 3.1).

A better choice for a sparse graphical inference would be, in most of the cases, to enforce the number of edges to be small, a function known as the ℓ_0-norm. Such norm is strongly non-convex. Generally, iterative algorithms strongly require a convex penalty for convergence guarantees, and for this reason the use of its convex relaxation, the ℓ_1-norm, is usually preferred. Recent work proceeds in the direction of using the ℓ_0-norm, based on the development of an algorithm for solving non-convex and non-smooth minimisation problems (Bolte, Sabach, and Teboulle, 2014; Geer and Bühlmann, 2013). The use of non-convex penalisation terms allows to overcome the bias introduced, for example, by the ℓ_1-norm for large coefficients (Wen et al., 2016).

Regularised methods for network inference, in particular exploiting the sparsity coming from the ℓ_1-norm, are widely used to estimate multiple networks at once (Guo et al., 2011; Honorio and Samaras, 2010; Kolar et al., 2010; Varoquaux et al., 2010; Xie, Liu, and Valdar, 2016). Indeed, the use of group lasso norms (ℓ_{21}) helps with the joint selection of features across multiple graphs. Other penalties can be exploited to enforce consistency between such graphs (Hallac et al., 2017). Section 2.4 formally introduces the multiple network inference problem, which serves as the link from static to dynamical network inference since they have similar formulations. A wide overview of the formalisation of temporal graphical models and relation with multiple regularised network inference methods is included in Section 2.5.

1.3 *Bayesian Structure Learning*

Methods for graphical inference are usually limited in the sense they provide a single graph as output. Indeed, a main challenge in the structure learning problem among a set of variables is the size of the hypothesis space, which includes up to $2^{d(d-1)/2}$ graphs on d variables (Moghaddam et al., 2009). Particularly in the case in which the available samples are much lower than the variables of the problem, it can be useful to adopt a Bayesian approach and infer a series of best graphs based on a posterior distribution, instead of relying on a single best graph (*i.e.*, a maximum likelihood estimation).

Probabilistic methods allow to assign a confidence interval to results, while efficiently dealing with noise by using appropriate priors on the problem at hand (Sanguinetti, Rattray, and Lawrence, 2006). Also, additional priors may be employed in order to deal with a small number of samples, for example by imposing a structure on the data. Hence, probabilistic methods are widely exploited to understand the relation between the variables, in particular for network inference problems (Pournara and Wernisch, 2007; Sabatti and James, 2005; Sanguinetti, Rattray, and Lawrence, 2006).

A major limitation in their use is the computation of the marginal likelihood associated to a general graphical model. Decomposable graphs (Definition A.2) allow a closed-form solution of the marginal likelihood (Dawid and Lauritzen, 1993). However, restricting to decomposable graphs is heavily limiting, since

the number of decomposable graphs is much less than the total number of general undirected graphs for a fixed number of variables (Murphy, 2012). Other authors considers the non-decomposable case (Atay-Kayis and Massam, 2005; Jones et al., 2005), but the methods are restricted to a small number of variables due to their use of expensive Monte Carlo approximations of the marginal likelihood. Lenkoski and Dobra (2008) proposed a Laplace approximation via an iterative proportional algorithm, that requires to compute the maximum a posteriori (MAP) estimate of the parameters under a G-Wishart prior (Roverato, 2002). Moghaddam et al. (2009) improve the MAP estimate, resulting in a faster method that do not need to know the cliques of the graph.

Such methods, however, still depend on the specification of a G-Wishart prior to efficiently compute the marginal likelihood, and the prior requires to know the graph underlying the variables. In practice, Moghaddam et al. (2009) rely on linear regression or lasso methods to identify the Markov blanket for each node, an approach similar to the graphical lasso (Section 2.3). Hence, resulting graphs are only sparse if the prior is sparse (*i.e.*, the graph specified in the prior is sparse) while the model does not embed a way to regulate the amount of sparsity of the graphs. Nonetheless, Bayesian methods are shown to improve maximum likelihood estimators in real contexts, even though their computational requirements are not optimal for large sets of variables.

Summary

This chapter briefly describes two general approaches for network inference. Due to their simplicity and effectiveness in real contexts, pairwise analysis is one of the best known approach for network inference. Another approach, instead, relies on a regularised inference, interpreting the task as a machine learning method which aims to find the best model describing available data.

Graphical models have received much attention due to their flexibility and the support of the theory. Indeed, the next chapter will continue with an extensive overview on graphical models for structure learning, which represent the foundations of the work of this thesis.

2 Gaussian Graphical Models

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{x_1, \dots, x_d\} = \{x_i\}_{i \in \Gamma}$ is a finite set of vertices, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of edges, a *graphical model* is a multivariate probability distribution on $x_1, \dots, x_d$ variables where the conditional independence between two variables x_i and x_j given all the others is encoded in $\mathcal{G}$ (Lauritzen, 1996). The two variables x_i and x_j are conditionally independent given the others if $(x_i, x_j) \notin \mathcal{E}$ and $(x_j, x_i) \notin \mathcal{E}$.

The focus of this thesis is restricted to undirected Gaussian graphical models (GGMs), where *(i)* there is no distinction between an edge $(x_i, x_j) \in \mathcal{E}$ and (x_j, x_i), and *(ii)* variables are jointly distributed according to a multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$. For simplicity, unless otherwise specified, throughout this thesis the normal distributions are assumed to be centred (without loss of generality), *i.e.*, $\boldsymbol{\mu} = \mathbf{0}$, thus depending only on the covariance matrix Σ (Choi et al., 2011). The inverse covariance matrix $\Theta = \Sigma^{-1}$, called *precision* matrix, encodes the conditional independence between pairs of variables, or, in other words, the structure of the graph. Indeed, the precision matrix has a zero entry in the position (i, j) (*i.e.*, $\Theta_{ij} = 0$) if and only if $(x_i, x_j) \notin \mathcal{E}$ (Lauritzen, 1996). For this reason, one can interpret the precision matrix as the weighted adjacency matrix of $\mathcal{G}$, encoding the dependence between variables. Hence, the study of the covariance matrix (and its inverse) is fundamental to understand the graphical model of variables which follow a multivariate normal distribution.

The problem of *graphical inference*, that aims at inferring the structure among observed variables starting from data, has received a lot of attention in recent years. In particular, this chapter presents two types of graphical inference, that is *(i) static* graphical inference, that aims at inferring the graphical model between a set of variables at a single time point (as in Sections 2.3 and 2.6), and *(ii) dynamical* graphical inference, that aims at inferring multiple graphical models for each time-points, exploiting the consistency of temporal states in a dynamical system (as in Section 2.5).

The goal of static methods is to infer a static network between variables. While powerful in real cases, such models do not consider relevant prior assumptions on the data, such as their inclusion into a global evolving system. Indeed, data coming from adjacent time points may direct the analysis for a more reliable inference of the graphical model between the variables at a particular time point, as exploited from time-varying models.

Outline

This chapter starts by recalling the continuous multivariate distributions which are fundamental for the understanding of the work of this thesis. In particular,

Section 2.1 presents the most common distribution in statistics and machine learning, namely the Gaussian (or normal) distribution, and clarifies the relation between a joint Gaussian distribution and the conditional independence between the variables. Section 2.2 presents the closely-related Wishart distribution, which is the distribution of positive-definite matrices. Then, this chapter continues with a list of the most known methods for the task of static graphical inference, starting from the *graphical lasso* (Section 2.3). Based on the graphical lasso, several variations have been proposed in the literature, accounting for more components in order to be able to capture the complexity of a wide range of systems. Examples include the *joint graphical lasso*, for multi-class graph inference (Section 2.4), the *time-varying graphical lasso*, for dynamical network inference (Section 2.5), and the *latent variables graphical lasso*, which considers the presence of latent unmeasurable factors during the inference of the network (Section 2.6). This chapter concludes with a method to use GGMs in presence of non-jointly Gaussian (but still continuous) variables, using the copula transformation (Section 2.7).

2.1 Gaussian Distribution

The multivariate Gaussian or multivariate normal is one the most important joint probability density function for continuous variables. Let $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ indicate a variable drawn from a multivariate normal, where $\boldsymbol{\mu} = \mathbb{E}[\boldsymbol{x}] \in \mathbb{R}^d$ is the mean vector, and $\Sigma = \text{cov}[\boldsymbol{x}] \in \mathcal{S}^d_{++}$ is the $d \times d$ covariance matrix. The probability density function (pdf) of a multivariate normal in d dimensions is defined as follows:

$$p(\boldsymbol{x}|\boldsymbol{\mu}, \Sigma) = (2\pi)^{-d/2}\det(\Sigma)^{-1/2} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right). \tag{2.1}$$

Consider a set of n independent and identically distributed (i.i.d.) samples in d dimensions, such that $X = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) \in \mathbb{R}^{n \times d}$ and $\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ for $i = 1, \dots, n$. Then, based on Equation (2.1) and using Equation (A.3), the Gaussian log-likelihood is as follows:

$$\begin{aligned}
\ell(\boldsymbol{\mu}, \Sigma) &= \log p(X|\boldsymbol{\mu}, \Sigma) = \log \prod_{i=1}^{n} p(\boldsymbol{x}_i|\boldsymbol{\mu}, \Sigma) \\
&= \log \prod_{i=1}^{n} (2\pi)^{-d/2} \det(\Sigma)^{-1/2} \exp\left(-\frac{1}{2}(\boldsymbol{x}_i-\boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu})\right) \\
&= \sum_{i=1}^{n} -\frac{d}{2}\log(2\pi) - \frac{1}{2}\text{logdet}(\Sigma) - \frac{1}{2}\left((\boldsymbol{x}_i-\boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu})\right) \\
&= -\frac{nd}{2}\log(2\pi) - \frac{n}{2}\text{logdet}(\Sigma) - \frac{1}{2}\left(\sum_{i=1}^{n}(\boldsymbol{x}_i-\boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu})\right) \\
&= -\frac{nd}{2}\log(2\pi) - \frac{n}{2}\text{logdet}(\Sigma) - \frac{n}{2}\text{tr}(S\Sigma^{-1}),
\end{aligned} \tag{2.2}$$

with $S = (1/n)\sum_{i=1}^{n}(\boldsymbol{x}_i-\boldsymbol{\mu})^\top(\boldsymbol{x}_i-\boldsymbol{\mu})$ covariance matrix.

Sometimes it is convenient to work in terms of the *precision* (or concentration) matrix $\Theta = \Sigma^{-1}$. An alternative parametrisation, based on the precision and empirical covariance matrices, leads to

$$\begin{aligned} \ell(S, \Theta) &= -\frac{nd}{2}\log(2\pi) + \frac{n}{2}\operatorname{logdet}(\Theta) - \frac{n}{2}\operatorname{tr}(S\Theta) \\ &\propto \operatorname{logdet}\Theta - \operatorname{tr}(S\Theta). \end{aligned} \tag{2.3}$$

Consider $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, *i.e.*, a random vector drawn from a multivariate normal distribution. Let the covariance Σ be *regular*, in the sense that the precision matrix $\Theta = \Sigma^{-1}$ is well defined. Then, the conditional independence between variables in the multivariate normal distribution is associated to zero entries in the precision matrix (Dempster, 1972).

Proposition 2.1. *Let $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, where Σ is regular. Let Γ be the set of entries in Σ. Then, for each $i, j \in \Gamma$ with $i \neq j$,*

$$x_i \perp\!\!\!\perp x_j \mid \boldsymbol{x}_{\Gamma\setminus\{i,j\}} \iff \theta_{ij} = 0 \tag{2.4}$$

where $\Theta = \{\theta_{ab}\}_{a,b\in\Gamma} = \Sigma^{-1}$ is the precision matrix of the distribution.

This result follows from standard linear algebra. Details and proof of the proposition can be found in (Lauritzen, 1996, Section 5.1.3). For this reason, the inverse covariance matrix is associated to the graph between the variables, where a link exists if and only if the two variables have a value different than zero in the corresponding entry of the precision matrix.

Markov properties on undirected graphs

Multivariate normal models defined by restricting particular elements in the inverse covariance matrices to be zero correspond to different Markov properties. Consider a general probability space $\mathcal{X}$, and cl and bd as defined in Appendix A.1. A probability measure P on $\mathcal{X}$, relative to a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, satisfies

(P) the pairwise Markov property, if for each pair $(a, b) \notin \mathcal{E}$

$$a \perp\!\!\!\perp b \mid \mathcal{V} \setminus \{a, b\},$$

(L) the local Markov property, if for any vertex $a \in \mathcal{V}$

$$a \perp\!\!\!\perp \mathcal{V} \setminus \operatorname{cl}(a) \mid \operatorname{bd}(a),$$

(G) the global Markov property, if for any triple (A, B, S) of disjoint subsets of $\mathcal{V}$ such that S separates A from B in $\mathcal{G}$

$$A \perp\!\!\!\perp B \mid S.$$

GGMs assume that the random vector $\boldsymbol{x}$ follows a multivariate normal distribution that satisfies the pairwise Markov property with respect to $\mathcal{G}$. Such density is continuous and positive, hence it implies global and local Markov properties (Lauritzen, 1996).

2.2 Wishart Distribution

The Wishart distribution is the sampling distribution of a positive definite matrix, with jointly Gaussian-distributed variables.

Definition 2.1 (Wishart distribution). *A random matrix $S \in \mathbb{R}^{d\times d}$ follows a d-dimensional Wishart distribution with parameter Σ and ν degrees of freedom if*

$$S = X^\top X, \tag{2.5}$$

where $X \in \mathcal{N}_{\nu\times d}(\mathbf{0}, I_\nu \otimes \Sigma)$. A matrix S which follows a Wishart distribution of dimension d is indicated with $S \sim \mathcal{W}_d(\nu, \Sigma)$.

In particular, the Wishart distribution defines a pdf over positive definite matrices S, as follows:

$$p(S|\nu, \Sigma) = \frac{\det(S)^{(\nu-d-1)/2}}{2^{\nu d/2}\det(\Sigma)^{\nu/2}\Gamma_d(\nu/2)} \exp\left(-\frac{1}{2}\operatorname{tr}(\Sigma^{-1}S)\right), \tag{2.6}$$

where $\Gamma_d(\cdot)$ is the multivariate Gamma function, defined as:

$$\Gamma_d(\nu/2) = \pi^{d(d-1)/4} \prod_{i=1}^{d} \Gamma\left((\nu + 1 - i)/2\right). \tag{2.7}$$

Remark. In the case where $d = 1$, the Wishart distribution reduces to the χ^2 distribution, *i.e.*, $\mathcal{W}_1(\nu, \sigma^2) = \sigma^2\chi^2(\nu)$.

Also, the Wishart distribution can be seen as the generalisation of the Gamma distribution to positive definite matrices. In particular, it can be used to model the uncertainty in covariance matrices Σ or their inverses Θ. It has relevant applications in Bayesian inference, as detailed in Chapter 8.

2.3 Graphical Lasso

Consider a series of samples drawn from a multivariate Gaussian distribution $X \sim \mathcal{N}(\mathbf{0}, \Sigma)$, $X \in \mathbb{R}^{n\times d}$. Network inference aims at recovering the graphical model of the d variables, *i.e.*, the interaction structure $\Theta = \Sigma^{-1}$ given n observed samples. The graphical modelling problem (in some cases also known as *covariance selection* problem) has been extensively tackled in the literature by estimating the precision matrix Θ instead of the covariance matrix Σ (Banerjee, Ghaoui, and d'Aspremont, 2008; Bien and Tibshirani, 2011; Friedman, Hastie, and Tibshirani, 2008; Lauritzen, 1996; Meinshausen and Bühlmann, 2006; Ravikumar et al., 2011; Wainwright, Ravikumar, and Lafferty, 2007; Yuan and Lin, 2007). This has been shown to improve the graphical model inference, particularly for high-dimensional problems (Meinshausen and Bühlmann, 2006). In such contexts, the assumption is that a variable is conditionally dependent only on a subset of all the others. Hence, a sparse prior may guide the estimation of the precision matrix in such a way to restrict the number of possible connections in the network, to improve interpretability and reduce

noise. Also, the imposition of a sparse prior on the problem helps with the identifiability of the graph, especially when the available number of samples is low compared to the dimensionality of the problem.

A sparse prior translates into forcing some connections in the graphical model to be zero, that are elements of the inverse covariance matrix (Proposition 2.1). As introduced in Section 1.2.2 this can be obtained using a ℓ_0-norm, which limits the number of non-zero components in the graph. The graphical inference problem can be interpreted as optimising the following functional:

$$\underset{\Theta}{\text{minimize}} \quad -\ell(S,\Theta), \qquad s.t. \ \|\Theta\|_{od,0} \leq k \tag{2.8}$$

where ℓ is the Gaussian log-likelihood (up to a constant and scaling factor) defined as $\ell(S,\Theta) = \text{logdet}(\Theta) - \text{tr}(S\Theta)$ for $\Theta \succ 0$ and $S = (1/n)X^\top X = (1/n)\sum_{i=1}^n \boldsymbol{x}_i \boldsymbol{x}_i^\top$ is the empirical covariance matrix. $\|\Theta\|_{od,0} = \sum_{i\neq j} \text{I}[\theta_{ij} \neq 0]$ is the ℓ_0-norm of the off-diagonal entries of Θ, which corresponds to the number of non-zero entries. The regularisation parameter k of the ℓ_0-norm limits the number of variables that can be (at most) different from 0.

The use of the ℓ_0-norm leads to an highly non-convex minimisation problem. Hence, it is usually better to use a convex relaxation, which translates into using a ℓ_1-norm. A model for the inference of Θ including the sparse (convex) prior is the *graphical lasso* (Friedman, Hastie, and Tibshirani, 2008; Hastie, Tibshirani, and Wainwright, 2015):

$$\underset{\Theta}{\text{minimize}} \quad -\ell(S,\Theta) + \alpha \|\Theta\|_{od,1}\,, \tag{2.9}$$

where $\|\cdot\|_{od,1}$ is the off-diagonal ℓ_1-norm, which promotes sparsity in the precision matrix (excluding the diagonal). Equation (2.9) has a lasso-like form (Tibshirani, 1996). In fact, the problem can be solved by coordinate descent, using a modified lasso regression on each variable in turn, thus leading to a simple, efficient and fast procedure.

Note that it is easy to modify the algorithm in order to have specific penalties α_{ik} for each edge. A value $\alpha_{ik} \longrightarrow \infty$ forces nodes x_i and x_j to be disconnected. This is particularly relevant in biology, where two variables (such as genes) are known not to interact directly.

The graphical lasso has solid theoretical guarantees, in particular consistency in the Frobenius norm (Rothman et al., 2008) and the operator-norm bound $\left\|\hat{\Theta} - \Theta^*\right\|_2$, where $\hat{\Theta}$ is the inferred network and Θ^* is the optimal one (Ravikumar et al., 2011).

A lasso-based neighbourhood selection for Gaussian graphical models was firstly proposed and developed by Meinshausen and Bühlmann (2006), with the following minimisation problem:

$$\hat{\theta}^a = \underset{\boldsymbol{\theta}:\theta_a=0}{\arg\min} \left\{ \frac{1}{n} \|X_a - X\boldsymbol{\theta}\|_2^2 + \alpha \|\boldsymbol{\theta}\|_1 \right\}, \tag{2.10}$$

with $\|\boldsymbol{\theta}\|_1 = \sum_i |\theta_i|$ is the ℓ_1-norm of the coefficient vector $\boldsymbol{\theta}$, and X_a corresponding to the n independent observations of the node a. For each node a, the

solution is found by minimising the coefficient on the edges of the neighbours of a. Neighbourhood selection with the lasso estimates the conditional independence separately for each node in the graph, iteratively. This is equivalent to variable selection for Gaussian linear models. From this perspective, the goal of the estimator is to find the zero-pattern of the inverse covariance matrix with the lasso procedure.

Consistency proofs of the estimator under high-dimensional scaling, as derived by Meinshausen and Bühlmann (2006), can be extended for logistic regression (Wainwright, Ravikumar, and Lafferty, 2007). Furthermore, Wainwright, Ravikumar, and Lafferty (2007) show how it is possible to establish sufficient conditions on the number of samples, dimensions and neighbourhood size to estimate the neighbourhood of each node simultaneously. On the contrary, this has been shown to be an approximation of the exact problem (Friedman, Hastie, and Tibshirani, 2008; Yuan and Lin, 2007). In particular, the neighbourhood selection with the lasso as proposed in (Meinshausen and Bühlmann, 2006) does not yield the maximum likelihood estimator when there is no equality between the empirical covariance matrix S (possibly perturbed by a matrix U) and the covariance estimated by the method. Friedman, Hastie, and Tibshirani (2008) bridge the conceptual gap between this and the exact problem proposing the graphical lasso method, based on the work of Banerjee, Ghaoui, and d'Aspremont (2008).

2.4 Multiple Network Inference

Consider the case in which samples belong to multiple classes, in such a way that their original covariance matrix depends from the particular class. Here, the assumption is that there is a covariance matrix $\Sigma_c = \Theta_c^{-1}$ for each class $c = 1, \ldots, C$. In other words, each sample is normally distributed according to the particular class it belongs to, *i.e.*,

$$p(\boldsymbol{x}|y = c, \theta) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_c, \Sigma_c). \tag{2.11}$$

This leads to the possibility of classifying a set of samples for which the distribution of the classes is known using the following decision rule, a technique called Gaussian discriminant analysis (GDA) (Murphy, 2012):

$$\hat{y}(\boldsymbol{x}) = \arg\max_c \ \log p(y = c|\boldsymbol{\pi}) + \log p(\boldsymbol{x}|\boldsymbol{\theta}_c), \tag{2.12}$$

where $\boldsymbol{\pi}$ is the prior on each class c, and $\boldsymbol{\theta}_c$ are the parameter of the distribution for class c. The posterior over class labels, using the definition of the Gaussian density, is as follows:

$$p(y = c|\boldsymbol{x}, \boldsymbol{\theta}) = \frac{\pi_c |2\pi\Sigma_c|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_c)^\top \Sigma_c^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_c)\right)}{\sum_{c'} \pi_{c'} |2\pi\Sigma_{c'}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_{c'})^\top \Sigma_{c'}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_{c'})\right)}, \tag{2.13}$$

which is known as quadratic discriminant analysis (QDA). Given a set of covariance matrices, it is possible to classify new samples based on the probability of belonging to a particular class.

A naïve learning method to infer the set of covariance (or precision) matrices would be the application of graphical lasso independently to each class. However, one may impose a certain consistency in the precision matrices of the C different classes. The addition of such constraint on problem (2.9) results in the *joint graphical lasso* (Danaher, Wang, and Witten, 2014):

$$\underset{(\Theta_1,\dots,\Theta_C)}{\text{minimize}} \sum_{i=1}^{C} \left[-n_i \ell(S_i, \Theta_i) + \alpha \left\|\Theta_i\right\|_{od,1}\right] + \beta\, P(\boldsymbol{\Theta}), \tag{2.14}$$

where P is a generic penalty imposed on the C precision matrices of the system. As in the rest of the thesis, without loss of generality the assumption is that each class is centred, so that $\boldsymbol{\mu}_c = \mathbf{0}$.

The idea is to impose a prior on multiple precision matrices, to limit the global behaviour of the system. Instead, one can interpret the classes as *time steps*, hence assigning a temporal ordering of the precision matrices. This idea will be the basis of the time-varying graphical lasso, detailed in the next section.

2.5 Time-Varying Network Inference

The analysis of a set of variables which describe the system at a particular time point is often little informative on the more global and general behaviour of the system. Consider the biological case, where genes interact with each other. Without further biological assumptions, static network inference answer to one possible question about the interaction of such genes. Furthermore, an instant later, one may ask the same question, and the static network inference method may answer again. By repeating this procedure, one can interpret the evolution of a system as a consecutive process of static network inference steps. However, two main problems remain.

Firstly, there is no theoretical guarantee that the network at step t would even be similar to the network at step $t + 1$, while one may intuitively expect so. In fact, there is no possibility to embed prior knowledge on the model on the evolution of the network.

Secondly, the presence of noise in particular time points of the network may conceal the global behaviour of the system, without a possibility to understand the changes due to the actual evolution of the system or to confounding factors. Indeed, the changes of the network at a particular time point may be due to external perturbation, noise or a particular developing state of the system.

Consider, as another practical example, a car which is making a right turn. Sensors associated to steering wheel, brake, velocity, gas pedal, etc., may offer information about how they are related to each other. However, only considering the state in which the car is in that particular moment (*i.e.*, making a turn) offers an explanation on the relations among the signals coming from the sensors on the machine, which will be supposedly different if the car changes its state (*e.g.*, going straight ahead or making a left turn).

Formally, problem (2.9) aims at recovering the structure of the system at fixed time (*static network inference*). However, complex systems have temporal dynamics that regulate their overall functioning (Albert, 2007; Friedman, Hastie, and Tibshirani, 2008). Hence, the modelling of such complex systems requires a *dynamical network inference*, where the states of the network are co-dependent. This naturally leads to the idea of *temporal consistency*, which assumes similarities between consecutive states of the network. In fact, one can assume that, for sufficiently close time points, a system shows negligible differences. During the inference of a dynamical network, temporal consistency may translate into the imposition of similarities among temporally close networks (Gibberd and Roy, 2017).

In particular, graphical lasso with temporal consistency results in *time-varying graphical lasso* (Hallac et al., 2017), where the inference of a network at a single time point t is guided by the states at adjacent time points.

Consider a series of observations $(\boldsymbol{x}^i(t))_{1\leq i\leq n_t}$, $\boldsymbol{x}^i(t) \in \mathbb{R}^d$, $t = 1, \dots, T$, each drawn from a multivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, \Sigma(t))$. Network inference aims at recovering at each time t the interaction structure $\Theta_t = \Sigma(t)^{-1}$ of the d variables, starting from n_t observations $\{\boldsymbol{x}^1(t), \dots, \boldsymbol{x}^{n_t}(t)\}$ (Hallac et al., 2017). Formally, let $\mathcal{S}^d_{++}$ be the convex cone of $d \times d$ positive-definite real matrices. The goal is to find a set of precision matrices $\Theta_t \in \mathcal{S}^d_{++}$, $t = 1, \dots, T$, that represents the dynamical network at different time points t. Then, the *time-varying graphical lasso* (TGL) problem is defined as follows:

$$\underset{\Theta_t \in \mathcal{S}^d_{++}}{\text{minimize}} \quad \sum_{t=1}^{T} -n_t \, \ell(S_t, \Theta_t) + \alpha \, \|\Theta_t\|_{\text{od},1} + \beta \sum_{t=1}^{T-1} \Psi(\Theta_{t+1} - \Theta_t), \tag{2.15}$$

where

- $S_t = (1/n_t) \sum_{i=1}^{n_t} \boldsymbol{x}^i(t) \otimes \boldsymbol{x}^i(t)$ is the empirical covariance matrix at time t;
- $\ell(S_t, \Theta_t) = \text{logdet}(\Theta_t) - \text{tr}(S_t\Theta_t)$ is the Gaussian log-likelihood (up to a constant and scaling factor), where Θ_t is positive definite;
- $\| \cdot \|_{od,1}$ is the off-diagonal ℓ_1-norm, which promotes sparsity in the precision matrix (excluding the diagonal);
- Ψ encodes prior information on the qualitative temporal behaviour of the network.

The penalty function Ψ and the parameter β specify the type of similarity imposed to consecutive time points and its strength, respectively. Such parameters can model a variety of behaviours. Options when choosing Ψ include the following (Hallac et al., 2017):

- Lasso penalty (ℓ_1) — $\Psi = \sum_{ij} |\cdot|$.
  Encourages few edges to change between subsequent time points, while the rest of the structure remains the same (Danaher, Wang, and Witten, 2014).

- Group lasso penalty ($\ell_1 2$) — $\Psi = \sum_j \| \cdot_j \|_2$.
  Encourages the graph to restructure at some time points and to stay stable in others (Gibberd and Roy, 2017; Hallac, Leskovec, and Boyd, 2015).
- Laplacian penalty (ℓ_2^2) — $\Psi = \sum_{ij} (\cdot_{ij})^2$.
  Encourages smooth transitions over time, for slow changes of the global structure (Weinberger et al., 2007).
- Max norm penalty (ℓ_∞) — $\Psi = \sum_j (\max_i | \cdot_{ij} |)$.
  Encourages a block of nodes to change their structure with no additional penalty with respect to the change of a single edge among such nodes. In fact, ℓ_∞ norm is influenced only from the most changing element for each row.
- Row-column overlap penalty — $\Psi = \min_{V : A = V + V^\top} \sum_j \|V_j\|_p$.
  Encourages a major change of the network at a specific time, while the rest of the system is enforced to remain constant. The choice of $p = 2$ causes the penalty to be node-based, *i.e.*, the penalty allows for a perturbation of a restricted number of nodes (Mohan et al., 2012).

Depending on prior assumptions on the problem, one may choose the most appropriate penalty for the data at hand.

A solution to problem (2.15) has been proposed via ADMM (Hallac et al., 2017). While favouring a relatively easy implementation for this model, ADMM requires a duplication of variables which may not always be feasible in practice, due to computational constraints, particularly for high-dimensional data.

2.6 Latent Variable Network Inference

Often, real-world observations do not conform exactly to a sparse GGM. This is due to global hidden factors that influence the system, which introduce spurious dependencies between observed variables (Choi, Chandrasekaran, and Willsky, 2010; Choi et al., 2011). For this reason, GGMs can be extended by introducing latent variables able to represent factors which are not observed in the data. These latent variables are *not* principal components, since they do not provide a low-rank approximation of the graphical model. On the contrary, such factors are added to the model in order to condition the statistics of the observed variables. In particular, one can consider both latent and observed variables to have a common domain (Choi et al., 2011).

Let latent variables be indexed by a set H of length h, and observed variables by a set O of length o. The precision matrix Θ of the joint distribution of both latent and observed variables may be partitioned into four blocks:

$$\Theta = \Sigma^{-1} = \left[\begin{array}{c|c} \Theta_H & \Theta_{HO} \\ \hline \Theta_{OH} & \Theta_O \end{array}\right],$$

so that $\Theta \in \mathbb{R}^{(h+o)\times(h+o)}$. Such blocks represent the conditional dependencies among latent variables (Θ_H), observed variables (Θ_O), between latent and observed (Θ_{HO}), and viceversa (Θ_{OH}). The marginal precision matrix Σ_O^{-1} of the observed variables is given by the Schur complement w.r.t. the block Θ_H (Chandrasekaran, Parrilo, and Willsky, 2010; Horn and Johnson, 2012):

$$\hat{\Theta}_O = \Sigma_O^{-1} = \Theta_O - \Theta_{OH}\Theta_H^{-1}\Theta_{HO}. \tag{2.16}$$

Θ_O specifies the precision matrix of the *conditional statistics* of the observed variables given the latent variables, while $\Theta_{OH}\Theta_H^{-1}\Theta_{HO}$ is a summary of the marginalisation effect over the latent variables. Such matrix has a small rank if the number of latent variables is small compared to the observed variables. Note that the rank is an indicator of the number of latent variables h (Chandrasekaran et al., 2011).

The effect of the marginalisation is scattered over many observed variables, in such a way not to confound it with the true underlying conditional sparse structure of Θ_O (Chandrasekaran, Parrilo, and Willsky, 2010). Typically $\hat{\Theta}_O$ is not sparse due to the low-rank term, while the addition of the latent factors contribution leads to recovering the true sparse GGM. For this reason, the graphical lasso in Equation (2.9) has been extended with the *latent variable graphical lasso* that includes the inference of a low-rank term, using the form (2.16), as follows (Chandrasekaran, Parrilo, and Willsky, 2010; Ma, Xue, and Zou, 2013):

$$\tilde{\Theta}, \tilde{L} = \underset{\substack{(\Theta, L) \\ L \succeq 0}}{\arg\min} \quad -\ell(S, \Theta - L) + \alpha\|\Theta\|_{od,1} + \tau\|L\|_*, \tag{2.17}$$

subject to $L \succeq 0$ and $\Theta - L \succ 0$ (this is required by the definition of ℓ, see Section 2.3). Here $\tilde{\Theta}$ provides an estimate of Θ_O (precision matrix of the observed variables) while $\tilde{L}$ provides an estimate of $\Theta_{OH}\Theta_H^{-1}\Theta_{HO}$ (marginalisation over the latent variables). Note that S is the empirical covariance matrix computed only on observed variables, since no information on the latent ones is available.

For completeness, Equation (2.18) introduces the original functional as presented in (Chandrasekaran, Parrilo, and Willsky, 2010), with a slightly different form:

$$\tilde{\Theta}, \tilde{L} = \underset{\substack{(\Theta, L) \\ L \succeq 0}}{\arg\min} \quad -\ell(S, \Theta - L) + \lambda(\gamma\|\Theta\|_1 + \|L\|_*), \tag{2.18}$$

which allows to prove that, with a suitable choice of λ, there exist a range of values of γ for which the estimates $(\hat{\Theta}, \hat{L})$ have, with high probability, the same sparsity and sign pattern and rank as $\Theta^*_{O,H}(\Theta^*_H)^{-1}\Theta^*_{H,O}$ (Chandrasekaran, Parrilo, and Willsky, 2010, Theorem 4.1). The following chapters of this thesis will rely on the form (2.17), which allows to model the two penalties (ℓ_1- and nuclear-norm) separately.

2.7 Copula for Non-Gaussian Graphical Models

GGMs assume variables to be jointly Gaussian. This assumption may not be satisfied in many applications, thus restricting the practical use of the graphical lasso and its derivations.

A simple generalisation to non-Gaussian (but still continuous) data exploits a *copula transformation*, which estimates a transformation from the empirical cumulative distribution function (cdf) of each variable such that the resulting data is jointly Gaussian (Liu, Lafferty, and Wasserman, 2009). Such transformation allows to compute the empirical covariance matrix corresponding to the Gaussian-transformed data and then apply the usual graphical lasso or its extensions (Liu et al., 2012).

Summary

Graphical models are widely studied in the literature, in particular during the last years, since the technology allows the analysis of a large number of variables and samples. In this context, Gaussian graphical models offer a great framework to model interactions between the variables in play. Starting from the graphical lasso, other methods have been developed to enhance the flexibility of the model, at the same time resulting in an increasing model complexity.

In particular, the assumption of latent factors is fundamental in real contexts where the information is partial, and data are conditioned on hidden components (Chandrasekaran, Parrilo, and Willsky, 2010). Also, the information coming from adjacent temporal states offers an efficient strategy to direct the analysis towards the inference of appropriate models, especially with limited data available. Indeed, such graphical models offered the basis of the work of this thesis, as detailed in Chapters 4 and 5.

3 Model Optimisation and Selection

The literature comprises numerous algorithms to optimise a so-called *objective function*, each with their respective advantages and drawbacks. Such algorithms aims to select the best model, given the data and possible constraints (called *priors* in different contexts).

A common trait of many machine learning methods (such as those described throughout this thesis, as in Chapters 4 and 5) and their corresponding algorithms is the presence of free parameters, which are usually called *hyper-parameters*. Such parameters must be specified before the actual learning step (*e.g.*, strength of the sparsity α or constraint on the latent variables τ for Equation (2.17)), as opposed to the model parameters, which are learned from data (*e.g.*, the interactions between variables). Complex models require the specification of hyper-parameters which directly affect the final model. In particular, distinct choices of a set of hyper-parameters correspond to different models. For this reason, the process of finding the best performing set of hyper-parameters is typically called *model selection*.

The choice of appropriate models for the data at hand must be supported by *(i)* reliable model assessment strategies and *(ii)* robust model selection techniques. First, a model assessment strategy estimates the generalisation error of the model on unseen data (Molinaro, Simon, and Pfeiffer, 2005). A particular model, instead, needs to be selected based on different criteria, searching across a set of possible models.

Outline

The rest of the chapter is organised as follows. Section 3.1 presents two general and widely used optimisation methods which serve as the basis for the work of this thesis, in particular for the time-varying graphical models in Chapters 4 and 5, namely ADMM (Section 3.1.1) and FBS (Section 3.1.2). Section 3.2 recalls commonly used model assessment procedures, exploited by the work of this thesis. Section 3.3 shows common model selection techniques, which aim to select the best model based on appropriate selection criteria. Section 3.4 concludes with different metrics for evaluating an inferred graphical model, based on the ground truth (in the case of synthetic data) or likelihood of the model on available data.

3.1 Optimisation Methods

Given a complex model, the goal of an optimisation algorithm is to find the solution to the problem (which, usually, has no closed-form) based on an iter-

ative process. The solution is usually interpreted as the value of the variables which minimise a given functional.

Indeed, functionals that lack a closed-form solution need to rely on optimisation methods. The choice of a particular optimisation method is usually based on constraints on the problem. This section focuses on two widely used optimisation methods, which allow for the minimisation of the functional described in the following chapters.

3.1.1 Alternating Direction Methods of Multipliers

The alternating direction methods of multipliers (ADMM) is a widely used optimisation method to solve problems which can be decomposed into smaller sub-problems, possibly subject to constraints (Boyd et al., 2010).

Consider an objective function of the following form:

$$\text{minimize} \quad f(\boldsymbol{x}) + g(\boldsymbol{z}), \tag{3.1}$$

subject to $A\boldsymbol{x} + B\boldsymbol{z} = \boldsymbol{c}$, with $\boldsymbol{x} \in \mathbb{R}^d$, $\boldsymbol{z} \in \mathbb{R}^e$, $A \in \mathbb{R}^{f \times d}$, $B \in \mathbb{R}^{f \times e}$, and $\boldsymbol{c} \in \mathbb{R}^f$. The optimal value of problem (3.1) is denoted by

$$p^* = \inf\{f(\boldsymbol{x}) + g(\boldsymbol{z}) | A\boldsymbol{x} + B\boldsymbol{z} = \boldsymbol{c}\}. \tag{3.2}$$

The augmented Lagragian for problem (3.1) is the following:

$$\mathcal{L}_\rho(\boldsymbol{x}, z, \boldsymbol{y}) = f(\boldsymbol{x}) + g(z) + \boldsymbol{y}^\top (A\boldsymbol{x} + Bz - \boldsymbol{c}) + \frac{\rho}{2} \|A\boldsymbol{x} + Bz - \boldsymbol{c}\|_2^2 . \tag{3.3}$$

A method of multipliers for problem (3.1) has the following form:

$$(\boldsymbol{x}^{k+1}, \boldsymbol{z}^{k+1}) = \arg\min_{\boldsymbol{x}, \boldsymbol{z}} \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}^k) \tag{3.4}$$

$$\boldsymbol{y}^{k+1} = \boldsymbol{y}^k + \rho(A\boldsymbol{x}^{k+1} + B\boldsymbol{z}^{k+1} - \boldsymbol{c}) \tag{3.5}$$

with $\rho > 0$. Here, the two variables in the augmented Lagrangian are jointly minimised. Instead, ADMM consists in the following iterations:

$$\boldsymbol{x}^{k+1} = \arg\min_{\boldsymbol{x}} \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{z}^k, \boldsymbol{y}^k) \tag{3.6}$$

$$\boldsymbol{z}^{k+1} = \arg\min_{\boldsymbol{z}} \mathcal{L}_\rho(\boldsymbol{x}^{k+1}, \boldsymbol{z}, \boldsymbol{y}^k) \tag{3.7}$$

$$\boldsymbol{y}^{k+1} = \boldsymbol{y}^k + \rho(A\boldsymbol{x}^{k+1} + B\boldsymbol{z}^{k+1} - \boldsymbol{c}), \tag{3.8}$$

where the two variables are updated in an alternating fashion, instead of a joint minimisation. See (Boyd et al., 2010) for details on the convergence of ADMM.

3.1.1.1 Scaled Form

In what follows, the ADMM will be prevalently used in its scaled form (since this leads to a simpler notation, but otherwise equivalent to an unscaled form) (Boyd et al., 2010). Let

$$\boldsymbol{r} = A\boldsymbol{x}^{k+1} + B\boldsymbol{z}^{k+1} - \boldsymbol{c}$$

be the residual. Then, from the Lagrangian in Equation (3.3),

$$\begin{aligned} \boldsymbol{y}^\top \boldsymbol{r} + \frac{\rho}{2} \|r\|_2^2 &= \frac{\rho}{2} \left\| \boldsymbol{r} + \frac{1}{\rho} \boldsymbol{y} \right\|_2^2 - \frac{1}{2\rho} \|\boldsymbol{y}\|_2^2 \\ &= \frac{\rho}{2} \|\boldsymbol{r} + \boldsymbol{u}\|_2^2 - \frac{\rho}{2} \|\boldsymbol{u}\|_2^2, \end{aligned}$$

where $\boldsymbol{u} = (1/\rho)\boldsymbol{y}$ is the scaled dual variable. With this rewriting, ADMM as in Equation (3.6) can be expressed as:

$$\boldsymbol{x}^{k+1} = \arg\min_{\boldsymbol{x}} \left(f(\boldsymbol{x}) + \frac{\rho}{2} \left\| A\boldsymbol{x} + B\boldsymbol{z}^k - \boldsymbol{c} + \boldsymbol{u}^k \right\|_2^2 \right) \tag{3.9}$$

$$\boldsymbol{z}^{k+1} = \arg\min_{\boldsymbol{z}} \left(g(\boldsymbol{z}) + \frac{\rho}{2} \left\| A\boldsymbol{x}^{k+1} + B\boldsymbol{z} - \boldsymbol{c} + \boldsymbol{u}^k \right\|_2^2 \right) \tag{3.10}$$

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^k + A\boldsymbol{x}^{k+1} + B\boldsymbol{z}^{k+1} - \boldsymbol{c}. \tag{3.11}$$

Hence, by defining $\boldsymbol{r}^k = A\boldsymbol{x}^k + B\boldsymbol{z}^k - \boldsymbol{c}$, it is clear that

$$\boldsymbol{u}^k = \boldsymbol{u}^0 + \sum_{j=1}^{k} r^j,$$

i.e., the running rum of residuals.

3.1.2 Forward-Backward Splitting

Forward-backward splitting (FBS) is an algorithm for the optimisation of objective functions of the following form (Combettes and Wajs, 2005):

$$\underset{x \in \mathcal{H}}{\text{minimize}} \quad f(x) + g(x), \tag{3.12}$$

where $\mathcal{H}$ is an Euclidean space, f is convex and smooth, while g is convex and possibly non-smooth. The idea behind the method is to make a descent step of size γ towards the direction of gradient of the smooth part f (the *forward* step), then project the point back via the proximity operator of g (the *backward* step), and finally to perform a relaxation step of size $\lambda \in]0, 1]$.

Such algorithm has strong theoretical guarantees (Beck and Teboulle, 2009; Combettes and Wajs, 2005). However, these results require the smooth part f to have a Lipschitz continuous gradient on the whole space $\mathcal{H}$. This is not the case in the context of Gaussian graphical models, since the negative Gaussian log-likelihood is defined on the open convex cone $\mathcal{S}_{++}^d \subset \mathbb{R}^{d \times d}$ and its gradient is not Lipschitz continuous (Banerjee, Ghaoui, and d'Aspremont, 2008).

Only recently, global convergence guarantees along with rates of convergence in function values were extended to a wider class of functions (Salzo, 2017) that indeed cover the objective problem in the case of time-varying graphical models (Chapter 4). Such guarantees rely on suitable line-search backtracking procedures that adaptively select the step-size γ and/or the relaxation parameter λ, keeping the iterations inside the domain $\mathcal{S}_{++}^d$. Algorithm 1

Algorithm 1: Forward-Backward splitting with Line Search (FBS-LS).

for $k = 1, \ldots$ **do**
- choose $\gamma_k \in \mathbb{R}_{++}$;
- $\hat{x}^k = x^k - \gamma_k \nabla f(x^k)$;
- $y^k = \text{prox}_{\gamma_k g}(\hat{x}^k) = \arg\min_x \gamma_k g(x) + \frac{1}{2}\|x - \hat{x}^k\|^2$;
- choose $\lambda_k \in]0, 1]$;
- $x^{k+1} = x^k + \lambda_k(y^k - x^k)$;

presents a generic form of FBS. For the sake of compactness let

$$J(x, \gamma, \lambda) = x + \lambda(\text{prox}_g(x - \gamma \nabla f(x)) - x),$$

so that $x^{k+1} = J(x^k, \gamma_k, \lambda_k)$.

LS(γ). Set $\lambda_k \equiv 1$ and let $\delta, \epsilon \in]0, 1[$, $\overline{\gamma} \in]0, 1]$. Then $\gamma_k = \bar{\gamma}\epsilon^i$ where i is the smallest integer so that

$$f(J(x^k, \gamma_k, 1)) - f(x^k) \leq \langle y^k - x^k | \nabla f(x^k) \rangle + \frac{\delta}{\gamma_k}\|y^k - x^k\|^2.$$

LS(γ, λ). Let $\delta, \epsilon \in]0, 1[$, $\overline{\gamma}, \overline{\lambda} \in]0, 1]$. Then $\gamma_k = \bar{\gamma}\epsilon^i$ and i is the smallest integer so that $y^k = \text{prox}_{\gamma_k g}(x^k - \gamma_k \nabla f(x^k)) \in \mathbb{S}^d_{++}$ and $\lambda_k = \bar{\lambda}\epsilon^j$ where j is the smallest integer so that

$$f(J(x^k, \gamma_k, \lambda_k)) - f(x^k) \leq \lambda_k \left(\langle y^k - x^k | \nabla f(x^k) \rangle + \frac{\delta}{\gamma_k}\|y^k - x^k\|^2 \right).$$

Salzo (2017) proved that Algorithm 1 with any of the above line searches gives a sequence $(x^k)_{k \in \mathbb{N}}$ converging to a minimiser of $f + g$ and such that $(f + g)(x^k) - \min_x(f + g)(x) = o(1/k)$.

3.1.2.1 Fixed-point Criterion

Fixed-point criterion for Algorithm 1 is defined as follows (Goldstein, Studer, and Baraniuk, 2014):

$$\begin{aligned}
x^* &= x^* + \lambda_k(\text{prox}_{\gamma_k g}(x^* - \gamma_k \nabla f(x^*)) - x^*) \\
0 &= \lambda_k(\text{prox}_{\gamma_k g}(x^* - \gamma_k \nabla f(x^*)) - x^*) \\
0 &= \text{prox}_{\gamma_k g}(x^* - \gamma_k \nabla f(x^*)) - x^* \\
0 &= x^* - \gamma_k \nabla f(x^*) - \gamma_k G(x^*) - x^* \\
0 &= -\gamma_k \nabla f(x^*) - \gamma_k G(x^*) \\
0 &= \nabla f(x^*) + G(x^*),
\end{aligned} \tag{3.13}$$

where $G \in \partial g(x^*)$ is a sub-gradient of g, and ∂g is the sub-differential of g. When g is differentiable, $G = \nabla g(x^*)$. The fixed-point property ensures that when the FBS is applied to an optimal point x^*, the forward descent step (the function f) moves the point to a new location, while the backward descent step (the function g) moves it back to x^* — hence, the line-search on λ has no effect when x^* is an optimal point.

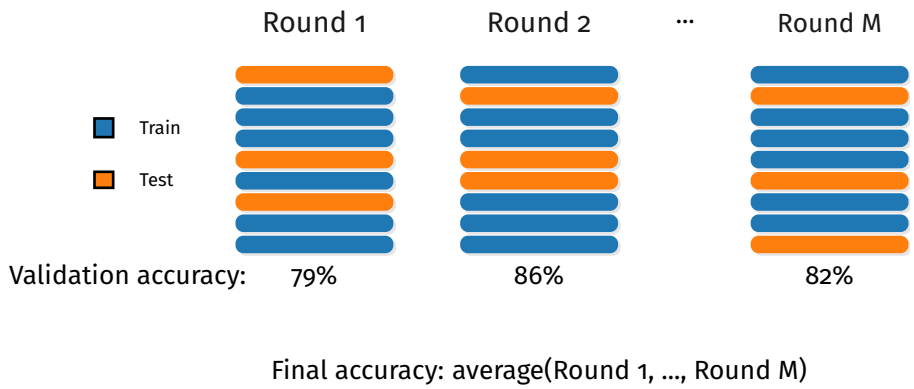


FIGURE 3.1. An example of MCCV. The dataset is divided into training and validation. A validation score is found as the average across the splits.

3.2 *Model Assessment*

Assessing a model performance translates into estimating its generalisation error, which measures the performance that the model is expected to achieve on future (*i.e.*, unseen) data. Since such data are, in general, not available, the predictive power of a learning machine should be evaluated on data simulating future observations. A popular class of techniques which simulate the acquisition of future data are resampling protocols (Molinaro, Simon, and Pfeiffer, 2005).

3.2.1 Monte Carlo Cross-Validation

The MCCV procedure repeatedly splits the n samples of the data set in two mutually exclusive sets. For each split, $n \cdot (1/v)$ samples are labelled as *validation* set and the remaining $n \cdot (1 - 1/v)$ as *training* set, with $v > 1$. The data points of the two sets are randomly sampled without replacement from the entire dataset. At each repetition, the learning machine is fitted on the training set and its predictive power is assessed by evaluating a performance metric (see Section 3.4) on the independent test set. Figure 3.1 shows an example of the MCCV.

3.2.2 k-fold Cross-Validation

The KFCV procedure splits the dataset into k non-overlapping subsets. At each iteration, one subset is kept aside as and used to estimate the prediction error of the model on the rest of the data. The final prediction error is the average on the k estimates obtained during the cross-validation procedure. In this cross-validation procedure, the number of splits k is limited by the number of samples at hand ($k \leq n$).

One advantage of a k-fold with respect to a MCCV strategy is a less intensive computational workload. Nonetheless, as the number of splits k cannot be

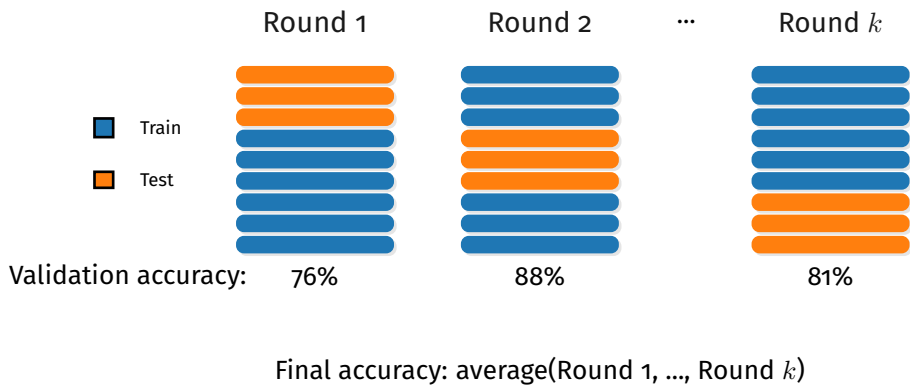


FIGURE 3.2. An example of KFCV. The dataset is divided into training and validation. A validation score is found as the average across the splits.

arbitrarily large, this method is not suitable for estimating the probability distribution of the generalisation error.

When $k = n$, the validation set has 1 sample and this strategy degenerates in the so-called leave-one-out cross-validation scheme. Figure 3.2 shows an example of the k-fold cross-validation.

3.3 *Model Selection*

Different techniques to perform hyper-parameters optimisation were presented over the years. While the base model may be the same, the choice of appropriate hyper-parameters for the data at hand is critical to define an appropriate and reliable model, to both generalise available samples and approximate unseen data.

Model selection strategies aim at generating a wide set of models based on a set of hyper-parameters. Then, the procedure selects the best model based on appropriate performance scores, or likelihood of the model on unseen data. Such strategies differ in how the choice of hyper-parameters is performed. Flexible machine learning models (such as the graphical lasso and its extensions) rely on a high number of hyper-parameters. This leads to computational challenges in the practical use of such models, due to the high number of possible hyper-parameter combinations that should be tested on the data. Hence, the specification of appropriate model selection procedures is fundamental for the use of complex models on both synthetic and real data.

3.3.1 Grid Search

A straightforward hyper-parameter optimisation strategy is the grid-search cross-validation. For a given model and cross-validation scheme, it generates a multi-dimensional grid of models based on a user-defined grid of hyper-parameters. This can be considered a brute-force algorithm, since evaluates every possible combination among the specified hyper-parameters. Hence, for

models that rely of an high number of hyper-parameters (such as the graphical models described in Chapters 4 and 5), this strategy can be computationally demanding even in presence of a restricted subspace for each hyper-parameter (due to the increase in the dimensions of the search space). In fact, such product over the sets makes grid search suffer from the *curse of dimensionality*, because the number of joint values grows exponentially with the number of hyper-parameters (Bergstra and Bengio, 2012).

Another drawback of this technique is the necessity to include the best combination of hyper-parameters in the search space. However, this may not happen in practice, and manual iterative refinements should be employed to focus the search on appropriate subsets of the search space.

3.3.2 Randomised Search

An alternative to standard grid-search cross-validation is its randomised version (Bergstra and Bengio, 2012). Instead of specifying a grid of hyper-parameters in advance, this strategy automatically generates a fixed number of models, each having hyper-parameters randomly sampled from given distributions. In other words, instead of specifying the *actual* hyper-parameters to test with the model (which in practice can be difficult), this strategy requires only the hyper-parameter *distributions*, and then explores the space following the distributions for each hyper-parameter.

Compared to an exhaustive search, this algorithm has two main benefits: *(i)* the possibility to specify a fixed budget (*e.g.*, a maximum number of models to generate) independent of the number of hyper-parameters and values, and *(ii)* the addition of hyper-parameters which do not influence the performance does not decrease the efficiency.

3.3.3 Bayesian Optimisation

A further step towards reducing the computational burden of model selection techniques, in particular oriented for complex and flexible models with a high number of hyper-parameters, is to employ an *active learning* procedure. In particular, the goal is to propose a meaningful and restricted subset of models among which to select the best one with respect to the available data.

Consider complex graphical models, such as those introduced in Chapter 2 or the temporal models contained in Chapters 4 and 5. Here, the number of possible combinations of hyper-parameters can be arbitrarily large. In order to avoid the assessment of a grid of models (which can be computationally expensive) and a random search on the hyper-parameters space, the idea of active learning is to interpret the scoring function as a learning task, and then to *propose* a new combination of hyper-parameters to both reduce the uncertainty in the unknown function and find the maximum value of the function (corresponding to the best model, given a performance score).

In this context, Bayesian optimisation has been shown to outperform state-of-the-art optimisation algorithms (Jones, 2001; Snoek, Larochelle, and Adams,

2012). Since performance scores associated to graphical models are continuous functions, it is possible to assume the unknown function to be sampled from a Gaussian process (GP) (Rasmussen, 2004). Then, the hyper-parameters may be sampled based on different strategies, such as the probability of improvement, expected improvement (EI) or GP upper confidence bound strategies (Snoek, Larochelle, and Adams, 2012).

The EI criterion, in particular, aims to maximise the expected improvement over the current best. Also, in the context of the GP, the EI strategy has a closed-form solution. The experiments of this thesis, unless otherwise specified, rely on a Bayesian optimisation technique for model selection based on the expected improvement strategy. Such strategy has shown to be better behaved than the probability of improvement, but unlike GP upper confidence bounds it does not require its additional tuning parameter, regulating the balance between exploitation against exploration.

3.4 Performance Metrics for Graphical Models

Particularly when dealing with real-world data sets, the learning of a model should be paired with a quantitative and robust performance assessment strategy. According to the learning task at hand different performance metrics may be used. For the assessment of a graphical model one should answer, at least, to the following questions:

(a) How much it is *likely* that new (*i.e.*, unseen) data come from such model?

(b) Does the model represent the real underlying structure of the system?

A simple answer to *(a)* might involve the likelihood (or the log-likelihood) of the model using Equation (2.2). The likelihood of the model does not require the ground truth, hence it is applicable to both synthetic and real data sets.

Instead, there can be different answers to *(b)*, depending on the performance metrics used. Such answers require the knowledge of the underlying system under analysis. It is possible to consider two different (but related) aspects in answering *(b)*. In particular, one can be interested in approximating the system where each weight of single edges influences the final score (*i.e.*, interpreting the structure learning as a regression problem), or approximating the structure of the system (*i.e.*, interpreting the structure learning as a classification problem).

Let $\Theta \in \mathcal{S}^d_{++}$ be the true graphical model, and $\hat{\Theta} \in \mathcal{S}^d_{++}$ the predicted one. For undirected graphical models, let only consider the upper triangular part of the graph, to avoid the duplication of the edges. Let $\Theta^{(u)}$ be the upper triangular part of Θ. Next sections aim at describing common ways to assess how distant the predictions $\hat{\boldsymbol{y}} = \hat{\Theta}^{(u)}$ are with respect to the actual output values $\boldsymbol{y} = \Theta^{(u)}$.

3.4.1 Structure Learning as Regression

A quantitative measure to assess the regression of the edges is the mean squared error (MSE), that incorporates bias and variance of the model. The

MSE of an unbiased estimator corresponds to its variance. Furthermore, the MSE is scale-dependent, defined as follows:

$$\mathrm{MSE}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2, \tag{3.14}$$

where $n = d(d-1)/2$. Hence, the MSE measures the distance between the entries of the inferred precision matrix with respect to the entries in the true underlying precision matrix.

3.4.2 Structure Learning as Classification

However, often one may be interested into the presence or absence of edges in the graph, disregarding their actual weight. In fact, for the same principles related to inferring a sparse network (Section 1.2.2), the presence or absence of edges in the graph plays an important role for the definition of interpretable models. In the context of graphical model assessment, it is possible to interpret the learning task as a binary classification problem, where classes consist in the presence or absence of an edge.

Formally, let *true/false positive* (TP/FP) be the number of correctly/incorrectly existing inferred edges, *true/false negative* (TN/FN) the number of correctly/incorrectly missing inferred edges (Hecker et al., 2009). In such way, the following classification metrics can be used for the assessment of a graphical model.

3.4.2.1 Accuracy

The accuracy of a model consists in the percentage of correct predictions with respect to the total number, defined as follows:

$$\mathrm{Acc}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(\hat{y}_i = y_i). \tag{3.15}$$

The best possible score is 1, while the expected score of a random classifier is the percentage of the most represented class over the total number of samples.

3.4.2.2 Balanced Accuracy

A drawback of the accuracy metric is that it is not easy to assess a model when the number of samples in each class is highly unbalanced. A simple extension which takes into account the number of samples in each class is the balanced accuracy score, for which a random classifier is constrained to return 0.5 independently from the number of samples in each class. The balanced accuracy is defined as follows:

$$\mathrm{Bacc}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{1}{2} \cdot \left[\frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{TN}} + \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \right]. \tag{3.16}$$

3.4.2.3 Precision

Also known as positive predictive value, the precision is the fraction of positive samples over the total number of samples classified as positive, defined as:

$$\mathrm{Prec}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}. \tag{3.17}$$

3.4.2.4 Recall

Also known as sensitivity or true positive rate (TPR), the recall measures the proportion of positive samples correctly classified as positive, defined as:

$$\mathrm{Rec}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{3.18}$$

3.4.2.5 F_1-score

The F_1-score is the harmonic mean of precision and recall, and it can be used to control both of them at the same time. The score is defined as:

$$\mathrm{F}_1(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FN} + \mathrm{FP}}. \tag{3.19}$$

3.4.2.6 True negative rate (TNR)

Also known as specificity, the TNR measures the proportion of negative samples which are classified as negative, thus including false positive samples. The score is defined as:

$$\mathrm{TNR}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}. \tag{3.20}$$

3.4.2.7 False positive rate (FPR)

Also known as fall-out, the FPR measures the proportion of negative samples which are incorrectly classified as positive, over all of the negative samples. The score is defined as:

$$\mathrm{FPR}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}. \tag{3.21}$$

3.4.2.8 False negative rate (FNR)

The FNR measures the proportion of positive samples which are incorrectly classified as negative, over all of the positive samples. The score is defined as:

$$\mathrm{FNR}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{\mathrm{FN}}{\mathrm{TP} + \mathrm{FN}}. \tag{3.22}$$

Summary

Efficient optimisation methods are required for a reliable inference of the optimal model based on the available data. ADMM is a powerful algorithm, which allow for the minimisation of complex functionals with a relatively easy implementation. However, when the functional increases in complexity and variables increase in size, the convergence of ADMM might not be optimal.

Section 3.1 presents both the widely used ADMM and advances on the FBS algorithm which allow for the minimisation of composite convex and possibly non-smooth objective functions (including sparsity-enforcing penalties). Both minimisation methods are exploited throughout this thesis, in particular for the development of the main contributions of this thesis (Chapters 4 and 5).

Such models depend on hyper-parameters, which require particular attention. Indeed, the choice of reliable models for a particular problem should be supported by appropriate model selection and assessment techniques, which in turn rely on different performance metrics.

Usually, there is no straightforward way to achieve this goal. Hence, it is important to assess an high number of models whenever possible, relying on performance scores which are appropriate for the task and take into account peculiarities of the data, such as possibly unbalanced classes of samples.

Part II

Time-Series Graphical Modelling

Recent advances on graphical models allow for their use with time-series data. Indeed, temporal consistency between samples can effectively be exploited during the inference of dynamical graphical models, as discussed in Section 2.5. This part includes the novel contributions of this thesis in the context of time-series graphical modelling, namely the time-varying graphical lasso with forward-backward splitting (Chapter 4) and the latent variable time-varying graphical lasso (Chapter 5). The work of Chapters 4 and 5 is included in (Tomasi et al., 2018a,b), respectively.

4 *Time-Varying Network Inference via Forward Backward Splitting*

Taking into account the dynamics of the system under analysis during the inference of a single graphical model may be beneficial in real use cases, helping in the case where the number of samples is lower than the number of variables. Since the system is assumed to follow a certain temporal behaviour, adjacent time points may help towards a reliable inference of the point-wise graphical model.

Motivation

While fundamental to represent real world data, the increasing complexity of temporal graphical models often challenges state-of-the-art minimisation methods, which need to deal with an increasing number of factors. This chapter focuses on the problem of graphical inference under a dynamical system, where a set of covariance matrices that describe the system is indexed by time, as introduced in Section 2.5. In particular, based on the time-varying graphical lasso, this chapter includes the first main contribution of this thesis, namely two algorithms for dynamical graphical models based on the forward-backward splitting minimisation procedure.

Such algorithms adopt two alternative line-search strategies that are studied in (Salzo, 2017). Considering the general Algorithm 1, in the first one the line search only involves the parameter γ_k, whereas the second method performs a line search on both γ_k and λ_k.

Outline

The rest of the chapter is organised as follows. Section 4.1 details the first main contribution of this thesis, providing two alternative FBS algorithms with line searches for two types of time-varying graphical lasso models. Section 4.2 contains an extensive validation of the FBS-based methods under synthetic experiments. Section 4.3 concludes with a discussion on the results.

4.1 Method

While powerful for real and complex systems, graphical models (in particular those for temporal data) introduce non-trivial challenges from a computational point of view. In this context, a popular optimisation algorithm is the alternating direction methods of multipliers (ADMM), which can cope with complex graphical lasso-based models (Boyd et al., 2010). ADMM partitions the problem

into multiple (easier) sub-problems, so that the solution of the global problem is found as the consensus among the solutions of the smaller sub-problems (Section 3.1.1). While offering a great flexibility for optimising complex models, a drawback of ADMM is the need of variable duplication before finding a consensus. This may lead to slow convergence rates and to a high computational cost (both in terms of computing resources and memory requirements). On the other hand, other optimisation algorithms make assumptions that are not satisfied in the setting of graphical models, thus (usually) not applicable in this context.

This section formally presents two problems of time-varying network inference under smooth and bounded variation temporal transitions. In particular, the time-varying graphical lasso is instantiated through specific evolutionary patterns (Section 2.5). The proposed methods cover two types of temporal transitions (Hallac et al., 2017): *(i)* a possibly discontinuous behaviour with few time changes in the links, by using a total variation penalty term; *(ii)* a smooth transition, adopting a square norm penalty term.

Hence, the first main contribution of this thesis consists in two procedures, based on the forward-backward splitting algorithm, that are ensured to converge to a (global) solution of the above problems. Such procedures rely on recent advances on the forward-backward splitting (FBS) method which introduces suitable line searches for the parameters of the algorithm and relax the assumptions, so to include the graphical modelling problem and its constraints (such as the positive-definitness of the precision matrix), while maintaining strong theoretical convergence guarantees (Salzo, 2017). Indeed, the two considered temporal transitions allow a separable form of the functionals that can be exploited by the FBS method, as shown in the following section.

The performance of the proposed algorithm is compared against the ground truth and the state-of-the-art minimisation algorithm in this context, that is, ADMM. The results show that the proposed FBS-based methods are significantly faster than ADMM. Also, since FBS algorithms do not require variable duplication, the spatial complexity is lower than ADMM, that is a fundamental feature for the analysis of large networks. This chapter aims at emphasising the need of investigating alternative optimisation methods for complex machine learning problems, which can deal with the increasing dimensionality of the data sets. This work is an attempt in this direction, showing FBS-based graphical models to be a effective alternative.

4.1.1 Problem Formulation

Consider the general form of FBS, as introduced in Section 3.1.2. The form (3.12) covers the *time-varying graphical lasso* (TGL) objective problem (2.15), where depending on the different choices of Ψ, the smooth part f may include the negative Gaussian log-likelihood only, or the last term too.

Different choices of Ψ reflect different evolutionary patterns of the interactions of variables in play and Hallac et al. (2017) consider several options. Among these consider the ℓ_1 norm, which gives rise to a total variation pen-

alty term, that is, $\Psi(\Theta_{t+1} - \Theta_t) = \|\Theta_{t+1} - \Theta_t\|_1 = \sum_{i,j=1}^{d} |\theta_{t+1,i,j} - \theta_{t,i,j}|$ and the square of the ℓ_2 norm, meaning that $\Psi(\Theta_{t+1} - \Theta_t) = \|\Theta_{t+1} - \Theta_t\|_2^2 = \sum_{i,j=1}^{d} |\theta_{t+1,i,j} - \theta_{t,i,j}|^2$. The first choice is suitable when one expects few edges to change between subsequent time points, whereas the second is appropriate when the dynamics smoothly varies over time (Hallac et al., 2017).

Consider the following two time-varying graphical lasso models:

$$\underset{\Theta_t \in \mathcal{S}_{++}^d}{\text{minimize}} \quad \sum_{t=1}^{T} -n_t \, \ell(S_t, \Theta_t) + \alpha \|\Theta_t\|_{\text{od},1} + \beta \sum_{t=1}^{T-1} \|\Theta_{t+1} - \Theta_t\|_1, \quad \text{(TGL-}\ell_1\text{)}$$

and

$$\underset{\Theta_t \in \mathcal{S}_{++}^d}{\text{minimize}} \quad \sum_{t=1}^{T} -n_t \, \ell(S_t, \Theta_t) + \alpha \|\Theta_t\|_{\text{od},1} + \beta \sum_{t=1}^{T-1} \|\Theta_{t+1} - \Theta_t\|_2^2, \quad \text{(TGL-}\ell_2^2\text{)}$$

where S_t is the empirical covariance matrix, defined as in Section 2.5.

In order to put the above minimisation problems in the form (3.12), let $\mathcal{H} = (\mathbb{R}^{d\times d})^T$, $\boldsymbol{\Theta} = (\Theta_t)_{1\le t\le T}$, with $\Theta_t = (\theta_{t,i,j})_{1\le i,j\le d} \in \mathbb{R}^{d\times d}$ being the precision matrix at time t, and consider f and g as follows:

Case (TGL-ℓ_1). $f(\boldsymbol{\Theta}) = \sum_{t=1}^{T} -n_t \, \ell(S_t, \Theta_t)$ and

$$\begin{aligned} g(\boldsymbol{\Theta}) &= \alpha \sum_{t=1}^{T} \|\Theta_t\|_{\text{od},1} + \beta \sum_{t=1}^{T} \|\Theta_{t+1} - \Theta_t\|_1 \\ &= \alpha \sum_{\substack{i,j=1 \\ i\neq j}}^{d} \sum_{t=1}^{T} |\theta_{t,i,j}| + \beta \sum_{i,j=1}^{d} \sum_{t=1}^{T} |\theta_{t+1,i,j} - \theta_{t,i,j}| \\ &= \sum_{i,j=1}^{d} \left[(1-\delta_{i,j}) \alpha \left\|\theta_{\cdot,i,j}\right\|_1 + \beta TV(\theta_{\cdot,i,j}) \right], \end{aligned}$$

where $\delta_{i,j}$ is the Kronecker symbol and $TV(\cdot)$ is the 1D total variation on $\mathbb{R}^T$.

Case (TGL-ℓ_2^2). $f(\boldsymbol{\Theta}) = \sum_{t=1}^{T} -n_t \, \ell(S_t, \Theta_t) + \beta \sum_{t=1}^{T-1} \|\Theta_{t+1} - \Theta_t\|_2^2$ and

$$g(\boldsymbol{\Theta}) = \alpha \sum_{t=1}^{T} \|\Theta_t\|_{\text{od},1} = \sum_{i,j=1}^{d} (1-\delta_{i,j}) \alpha \left\|\theta_{\cdot,i,j}\right\|_1 .$$

In both such cases the function g is convex and separable, meaning that

$$g(\boldsymbol{\Theta}) = \sum_{i,j=1}^{d} g_{i,j}(\boldsymbol{\theta}_{i,j}), \quad \boldsymbol{\theta}_{i,j} = (\theta_{t,i,j})_{1\le t\le T},$$

where, for every $(i,j) \in \{1,\dots,d\}^2$,

$$g_{i,j} \colon \mathbb{R}^T \to \mathbb{R}, \quad g_{i,j}(\boldsymbol{\theta}) = \begin{cases} (1-\delta_{i,j}) \alpha \|\boldsymbol{\theta}\|_1 + \beta TV(\boldsymbol{\theta}) & \text{in the case (TGL-}\ell_1\text{)}, \\ (1-\delta_{i,j}) \alpha \|\boldsymbol{\theta}\|_1 & \text{in the case (TGL-}\ell_2^2\text{)}. \end{cases}$$

The proximity operator of g can be computed component-wise (Combettes and Wajs, 2005). Moreover, in the case (TGL-ℓ_1) the components of g consist in a 1D total variation penalty if $i = j$ and in a fused lasso penalty otherwise. It is possible to exactly compute the proximity operator of such penalties by means of a finite termination procedure (Condat, 2013).

In the case (TGL-ℓ_2^2) the proximity operator of $g_{i,j}$ reduces to a soft-thresholding operation (Combettes and Wajs, 2005). For each of the above cases, the function f is defined on the open convex cone $\mathcal{S}_{++}^d$, it is convex, and its gradient is

$$\nabla f(\boldsymbol{\Theta}) = \begin{cases} \begin{pmatrix} n_1(S_1 - \Theta_1^{-1}) \\ n_2(S_2 - \Theta_2^{-1}) \\ \cdots\cdots \\ n_T(S_T - \Theta_T^{-1}) \end{pmatrix} & \text{in the case (TGL-}\ell_1\text{)} \\ \begin{pmatrix} n_1(S_1 - \Theta_1^{-1}) \\ n_1(S_2 - \Theta_2^{-1}) \\ \cdots\cdots \\ n_T(S_T - \Theta_T^{-1}) \end{pmatrix} + 2\beta \begin{pmatrix} \Theta_1 - \Theta_2 \\ 2\Theta_2 - \Theta_1 - \Theta_3 \\ \cdots\cdots \\ \Theta_T - \Theta_{T-1} \end{pmatrix} & \text{in the case (TGL-}\ell_2^2\text{).} \end{cases} \tag{4.1}$$

This shows that ∇f is (only) locally Lipschitz continuous on $\mathcal{S}_{++}^d$.

4.1.2 Algorithm

This section presents two instances of Algorithm 1 which correspond to two types of line-search procedures and can be applied to both problems (TGL-ℓ_1) and (TGL-ℓ_2^2).

Algorithm 2 implements a line-search on the step-size γ only, whereas Algorithm 3 performs a line-search on the relaxation parameter λ and an additional backtracking procedure on the step-size γ to keep the sequence of the iterates feasible. The operations $\text{prox}_{\gamma g_{i,j}}$ and ∇f are those described in Section 4.1.1.

Since f and g are convex and ∇f is locally Lipschitz continuous on $\mathcal{S}_{++}^d$, it follows from (Salzo, 2017) that Algorithms 2 and 3 are ensured to converge with a rate $o(1/k)$. Both the proposed algorithms are equivalent in terms of convergence properties and computational cost. However, in practice, they may behave differently depending on the applications, as shown in Section 4.2.

4.1.2.1 Stopping Criterion

Since $Y^k = \text{prox}_{\gamma_k g}(\hat{\boldsymbol{\Theta}}^k)$, it follows that $(\hat{\boldsymbol{\Theta}}^k - Y^k)/\gamma_k \in \partial g(Y^k)$, where ∂g is the subdifferential of g. The stopping criterion is based on the following residual

$$R^k = \nabla f(Y^k) + \frac{\hat{\boldsymbol{\Theta}}^k - Y^k}{\gamma_k}, \tag{4.2}$$

Algorithm 2: FBS-LS(γ) for time-varying network inference.

choose $\epsilon \in]0, 1[$, $\bar{\gamma} > 0$, and $\delta \in]0, 1[$;
choose $\boldsymbol{\Theta}^0$ and set $\gamma_{-1} = \bar{\gamma}\epsilon$;
for $k = 0, 1, \ldots$ *(until convergence)* **do**
  initialise $\gamma = \gamma_{k-1}/\epsilon$;
  do
    $\gamma \leftarrow \gamma\epsilon$;
    $\hat{\boldsymbol{\Theta}}^k = \boldsymbol{\Theta}^k - \gamma \nabla f(\boldsymbol{\Theta}^k) = (\hat{\boldsymbol{\theta}}^k_{ij})_{1 \le i,j \le d}$;
    for *each interaction* ij **do**
      $\boldsymbol{y}^k_{i,j} = \mathrm{prox}_{\gamma g_{i,j}}(\hat{\boldsymbol{\theta}}^k_{ij})$;
    $\boldsymbol{Y}^k = (\boldsymbol{y}^k_{i,j})_{1 \le i,j, \le d} = (Y^k_1, \ldots, Y^k_T)$;
    $\zeta_k = \langle \boldsymbol{Y}^k - \boldsymbol{\Theta}^k, \nabla f(\boldsymbol{\Theta}^k) \rangle + (\delta/\gamma) \left\| \boldsymbol{Y}^k - \boldsymbol{\Theta}^k \right\|_2^2$;
  while $Y^k_1 \not\succ 0$ *or* $Y^k_2 \not\succ 0, \cdots,$ *or* $Y^k_T \not\succ 0$ *or* $f(\boldsymbol{Y}^k) - f(\boldsymbol{\Theta}^k) > \zeta_k$;
  $\gamma_k = \gamma$;
  $\boldsymbol{\Theta}^{k+1} = \boldsymbol{Y}^k$;

which belongs to $\partial(f + g)(\boldsymbol{Y}^k)$ and, in view of the expression of $\hat{\boldsymbol{\Theta}}$ in Algorithms 2 and 3, can also be written as

$$\boldsymbol{R}^k = \nabla f(\boldsymbol{Y}^k) - \nabla f(\boldsymbol{\Theta}^k) + \frac{\boldsymbol{\Theta}^k - \boldsymbol{Y}^k}{\gamma_k}. \tag{4.3}$$

Since $\boldsymbol{Y}^k - \boldsymbol{\Theta}^k \to 0$ as $k \to +\infty$ (Salzo, 2017) and ∇f is locally Lipschitz continuous, it follows that $(\boldsymbol{R}^k)_{k \in \mathbb{N}}$ is a sequence of subgradients of $f + g$ (each one at the point $\boldsymbol{Y}^k$), that converges to zero. A scale invariant stopping criterion can be obtained by adopting the condition $r^k_r < \epsilon_{\text{abs}}$ or $r^k_n < \epsilon_{\text{abs}}$, where

$$r^k_r = \frac{\left\| \boldsymbol{R}^k \right\|_2}{\max\left\{ \left\| \nabla f(\boldsymbol{Y}^k) \right\|_2, \left\| \frac{\hat{\boldsymbol{\Theta}}^k - \boldsymbol{Y}^k}{\boldsymbol{Y}^k} \right\|_2 \right\} + \epsilon_r} \qquad \text{and} \qquad r^k_n = \frac{\left\| \boldsymbol{R}^k \right\|_2}{\left\| \boldsymbol{R}^1 \right\|_2 + \epsilon_n},$$

are the *relative* and *normalised* residuals respectively, ϵ_{abs} is an absolute tolerance parameter, and ϵ_r and ϵ_n are small constants to prevent the denominator from being zero (Goldstein, Studer, and Baraniuk, 2014).

4.1.2.2 Complexity

In Algorithms 2 and 3 the most expensive step lies in the inversion of T matrices, required by the gradient of f, as in Equation (4.1). The complexity per iteration is equivalent to that of the ADMM proposed by (Hallac et al., 2017).

Indeed, the optimisation of TGL with ADMM requires an eigenvalue decomposition at each iteration, so both minimisation methods have computational

Algorithm 3: FBS-LS(γ, λ) for time-varying network inference.

choose $\epsilon \in]0, 1[$, $\bar{\gamma} > 0, \bar{\lambda} \in]0, 1]$, and $\delta \in]0, 1[$;
choose $\boldsymbol{\Theta}^0$ and set $\gamma_{-1} = \bar{\gamma}\epsilon, \lambda_0 = \bar{\lambda}$;
for $k = 0, 1, \ldots$ *(until convergence)* **do**
  initialise $\gamma = \gamma_{k-1}/\epsilon$ and $\lambda = \lambda_{k-1}/\epsilon$;
  do
    $\gamma \leftarrow \gamma\epsilon$;
    $\hat{\boldsymbol{\Theta}}^k = \boldsymbol{\Theta}^k - \gamma \nabla f(\boldsymbol{\Theta}^k) = (\hat{\boldsymbol{\theta}}^k_{ij})_{1 \le i,j \le d}$;
    for *each interaction* ij **do**
      $\boldsymbol{y}^k_{i,j} = \text{prox}_{\gamma g_{i,j}}(\hat{\boldsymbol{\theta}}^k_{ij})$;
    $\boldsymbol{Y}^k = (\boldsymbol{y}^k_{i,j})_{1 \le i,j, \le d} = (Y^k_1, \ldots, Y^k_T)$;
  while $Y^k_1 \not\succ 0$ *or* $Y^k_2 \not\succ 0, \cdots,$ *or* $Y^k_T \not\succ 0$;
  do
    $\lambda \leftarrow \lambda\epsilon$;
    $\boldsymbol{\Theta}^{k+1} = \boldsymbol{\Theta}^k + \lambda(\boldsymbol{Y}^k - \boldsymbol{\Theta}^k)$;
    $\zeta_k = \langle \boldsymbol{Y}^k - \boldsymbol{\Theta}^k, \nabla f(\boldsymbol{\Theta}^k) \rangle + (\delta/\gamma) \left\| \boldsymbol{Y}^k - \boldsymbol{\Theta}^k \right\|_2^2$;
  while $f(\boldsymbol{\Theta}^{k+1}) - f(\boldsymbol{\Theta}^k) > \lambda\zeta_k$;
  $\gamma_k = \gamma$;

complexity of $O(Td^3)$ per iteration. However, both the matrix inversion problem and the eigenvalue decomposition problem can be solved by more efficient algorithms with lower complexity. Employing such algorithms can significantly improve the time performance of the proposed FBS-based algorithms as well as of ADMM.

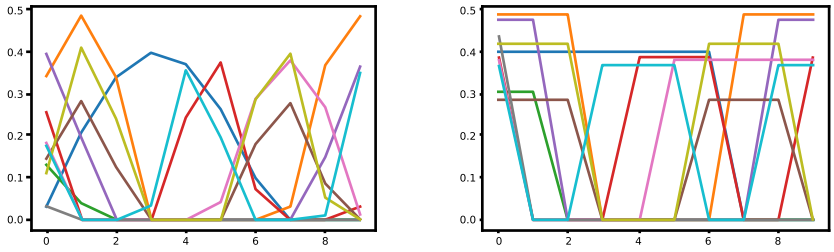
Finally, exploiting a particular structure of the $\boldsymbol{\Theta}$ matrix (such as a block structure) may be an additional benefit in the matrix inversion operation. Such improvements are left as a future work.

4.2 *Experiments*

The performance of the proposed methods has been assessed on synthetic data in terms of the number of iterations, execution time, and space scalability. In particular, the two proposed algorithms FBS-LS(γ) (Algorithm 2) and FBS-LS(γ, λ) (Algorithm 3) are compared to ADMM, which is the state of the art in the context of time-varying network inference (Hallac et al., 2017).

4.2.1 Convergence

Data were generated starting from a set of precision matrices $\boldsymbol{\Theta} = (\Theta_1, \ldots, \Theta_T)$, related in time according to a specific behaviour while guaranteeing that $\Theta_t \in \mathcal{S}^d_{++}$ for $t = 1, \ldots, T$. In particular, the data sets considered consisted of $n_t = 200$ samples in $\mathbb{R}^d$ with $d = 200$ and $T = 10$ time stamps. Such data set follow two different temporal behaviours.



(A) Smooth signal. (B) Square signal.

FIGURE 4.1. Example of the smooth and square signals used to generate the synthetic data sets.

	precision	0.1		0.01		0.001	
	score	iter.	time [s]	iter.	time [s]	iter.	time [s]
ℓ_1	FBS-LS(γ)	22 ± 1	4.8 ± 0.7	24 ± 1	5.1 ± 0.4	26 ± 1	5.0 ± 0.3
	FBS-LS(γ,λ)	22 ± 1	4.6 ± 0.7	24 ± 1	5.4 ± 0.5	26 ± 1	5.6 ± 0.5
	ADMM	1060 ± 553	75.2 ± 39.8	2623 ± 1757	184.3 ± 122.3	4312 ± 1536	301.6 ± 107.4
ℓ_2^2	FBS-LS(γ)	72 ± 19	7.9 ± 2.7	107 ± 30	11.7 ± 4.1	137 ± 40	14.9 ± 5.5
	FBS-LS(γ, λ)	72 ± 19	8.3 ± 2.8	104 ± 31	12.1 ± 4.5	129 ± 41	14.9 ± 6.0
	ADMM	192 ± 24	13.2 ± 1.7	252 ± 42	17.3 ± 3.1	453 ± 66	30.4 ± 3.8

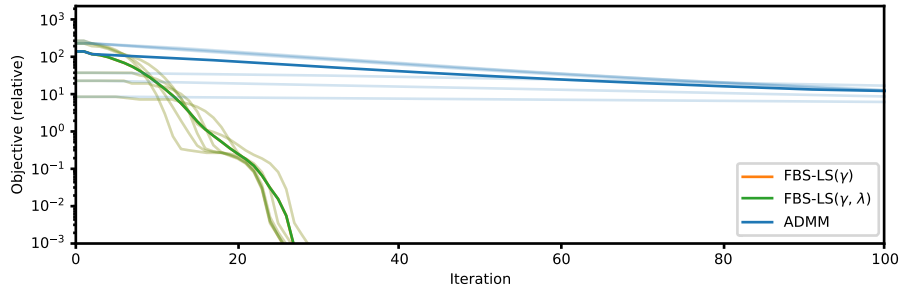
TABLE 4.1. Comparison between FBS with line search and ADMM. The algorithms were employed with several values of (α, β). The table displays the average and standard deviation of the number of iterations and CPU times across the different runs for achieving $|\text{obj}_k - m_*|/|m_*| \leq \varepsilon$, with $\varepsilon \in \{0.1, 0.01, 0.001\}$. For each pair of hyper-parameters, the minimum m_* is estimated as the best value obtained in 500 iterations among the different algorithms.

For the first data set, interactions between variables across time follow a square waveform behaviour. Under such schema, the interactions may be zero or positive at particular time points, but the transition between those states is non-smooth.

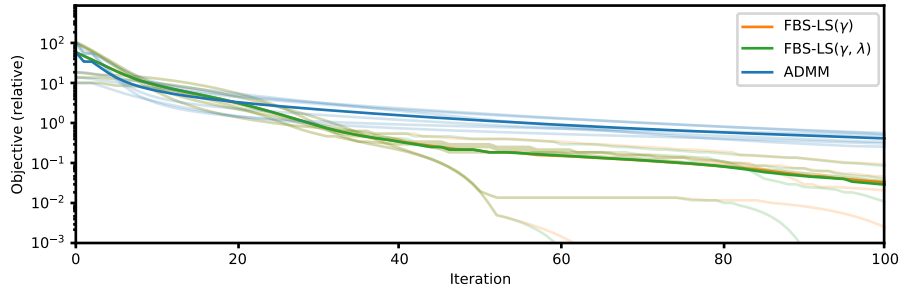
For the second data set, variable interactions follow to a smooth sinusoidal behaviour, hence changing slowly in time. Additional details on data generation can be found in the implementation (see Availability and Implementation).

The experiments include the time-varying graphical lasso with the two temporal penalties (TGL-ℓ_1) and (TGL-ℓ_2^2), according to the type of the data set. As for the hyper-parameters (α, β), the search space was $[0.1, 1]\times[0.1, 5]$ for (TGL-ℓ_1) and $[0.1, 1] \times [0.01, 0.1]$ for (TGL-ℓ_2^2). A Bayesian optimisation procedure ensured that the best hyper-parameters lie in the interior of the search space (do not belong to the boundary). In particular, $(\alpha^*, \beta^*) = (0.111, 4.855)$ for (TGL-ℓ_1), while $(\alpha^*, \beta^*) = (0.789, 0.020)$ for (TGL-ℓ_2^2). Then, I set a grid on the search space and ran the two proposed algorithms FBS-LS(γ) and FBS-LS(γ, λ) as well as ADMM, for the corresponding values of the hyper-parameters.

The performance of the proposed methods was evaluated with respect to the ground truth in terms of the mean squared error (MSE) for each algorithm



(A) ℓ_1-norm temporal penalty.



(B) ℓ_2^2-norm temporal penalty.

FIGURE 4.2. Relative objective value (decreasing) at each iteration. The relative value is obtained as $|\text{obj}_k - m_*|/|m_*|$, where m_* is the minimum objective value obtained across 500 iterations, and obj_k is the value of the objective function at iteration k. In both cases, FBS-based algorithms converge to the minimum faster with respect to ADMM.

after convergence, defined as:

$$\text{MSE} = \frac{2}{Td(d-1)} \sum_{t=1}^{T} \|\Theta_t^{(u)} - \tilde{\Theta}_t^{(u)}\|_2^2,$$

where Θ_t denotes the ground truth, $\tilde{\Theta}_t$ is the inferred precision matrix, and the superscript (u) refers to the upper triangular part of the matrix (excluding the diagonal). The achieved MSE was the same for each algorithm ($0.648 \cdot 10^{-4}$ for (TGL-ℓ_1), $0.498 \cdot 10^{-4}$ for (TGL-ℓ_2^2)).

Table 4.1 reports the performance of the three algorithms across the different runs in terms of the number of iterations and CPU times for achieving a given precision. Indeed, FBS- and ADMM-based methods have different stopping criterions. Hence, the comparison of such methods is based on their convergence with respect to an empirical minimum value of the objective function. In particular, for each combination of hyper-parameters, the methods ran for a fixed number of iterations (500) and the best objective value was selected across all methods. Then, both FBS- and ADMM-based methods were run again using as stopping criterion the closeness of the current objective value with the best one previously selected for the particular combination of hyper-parameters, with different precisions (namely, 0.1, 0.01 and 0.001). In

this experiment, both FBS-based algorithms outperformed ADMM. FBS-based algorithms were able, in only a few iterations, to increase the precision of order of magnitudes for both ℓ_1 and ℓ_2^2 set of experiments. The difference in the convergence behaviour with respect to ADMM was less substantial in the case of ℓ_2^2. In the case of ℓ_1, FBS has a higher cost per iteration with respect to ADMM. This is due to the computation of the proximity operator of the fused lasso penalty.

In the case of ℓ_2^2, instead, the cost is lower because the proximity operator of the nonsmooth (penalty) term simplifies to a soft-thresholding. Finally, for (TGL-ℓ_2^2), I point out the better performance of FBS-LS(γ, λ) against FBS-LS(γ).

Figure 4.2 shows the relative objective value across the first 100 iterations and multiple runs for FBS-based algorithms and ADMM. The averaged value is depicted in bold line. In particular, in the case of the (TGL-ℓ_1), FBS-based algorithms clearly surpass the ADMM in terms of convergence rate (Figure 4.2A).

The two algorithms FBS-LS(γ) and FBS-LS(γ, λ) overlap in the case of (TGL-ℓ_1), whereas FBS-LS(γ, λ) shows to convergence slightly faster than FBS-LS(γ). The poor convergence rate of ADMM may be due to the need of reaching a consensus among a large number of variables which is a typical scenario in the inference of time-varying networks.

4.2.2 Scalability

FBS- and ADMM-based optimisations feature different memory requirements. In particular, FBS-based implementation requires $O(2d^2T)$ in space, for keeping in memory both the precision and empirical covariance matrices at all time points. Instead, ADMM-based implementation requires more variables due to the consensus framework and the presence of dual variables. More specifically, in such setting, ADMM requires $O(4d^2(2T-1))$ space complexity (Hallac et al., 2017). The difference between the two complexities consists in a multiplicative factor which impact the analysis of large data sets.

Figure 4.3 shows the difference in space complexity as the number of unknowns ($Td(d+1)/2$) of the problem grows. Such computations do not take into account optimised data structures for sparse data. Exploiting the structure and the sparsity of the involved matrices may lead to better computational efficiency, but such investigation are left for future work.

4.2.3 Model Selection

The hyper-parameters of the methods have been selected by using a MCCV producedure (Section 3.2.1), based on the average maximum likelihood of the model across multiple splits of the data set. The selected hyper-parameters were used for both FBS- and ADMM-based algorithms, for which the functional is the same. Since the number of TGL-FB hyper-parameters is large, the model was selected according to a Gaussian process-based Bayesian optimisation procedure, based on the expected improvement strategy (Snoek, Larochelle, and Adams, 2012).

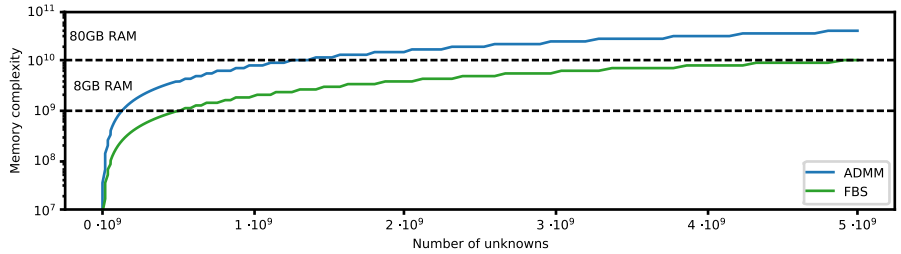


FIGURE 4.3. Memory requirements as the number of unknowns grows, with $T = 50$ and d varying. Each matrix entry is stored in double precision.

Note that, when the temporal dynamics is not known, it is possible and advantageous to include the choice of the temporal behaviour in a cross-validation procedure.

4.3 Discussion

This chapter proposes two novel algorithms for the graphical modelling of multivariate time-series, based on a forward-backward splitting procedure under mild-differentiability assumptions. Such algorithm covers two significant types of temporal behaviours, that is, a nonsmooth transition with few interaction changes, and a smooth transition with slow changes of the global system. Optimisation algorithms which are usually exploited for network inference suffer from drawbacks when considering large sets of unknowns. The experiments in this chapter proved that the proposed method is more efficient than ADMM for number of iterations, CPU time and memory requirements.

The possibility of applying the FBS algorithm on different graphical models, such as those considered in (Danaher, Wang, and Witten, 2014; Tomasi et al., 2018b), opens the way to the use of graphical models in large-scale data. When increasing the complexity of models, FBS-based graphical models would prove even more effective for real-world applications. Indeed, particularly for increasing data sets and model complexity, fast and theoretically sound optimisation algorithms represent a necessary tool for the graphical modelling community. In such context, this work could pave the way to develop solid graphical models with increasing complexity, leading to further advances in pattern recognition.

5 *Latent Variable Time-Varying Network Inference*

In many applications of finance, biology and sociology, complex systems involve entities interacting with each other. These processes have the peculiarity of evolving over time and of comprising latent factors, which influence the system without being explicitly measured. This chapter present a second main contribution of this thesis, that is the *latent variable time-varying graphical lasso* (LTGL), a method for multivariate time-series graphical modelling that considers the influence of hidden or unmeasurable factors. The estimation of the contribution of the latent factors is embedded in the model which produces both sparse and low-rank components for each time point. In particular, the first component represents the connectivity structure of observable variables of the system, while the second represents the influence of hidden factors, assumed to be few with respect to the observed variables. The LTGL model includes temporal consistency on both components, providing an accurate evolutionary pattern of the system.

In what follows, I will derive a tractable optimisation algorithm based on alternating direction method of multipliers, and develop a scalable and efficient implementation which exploits proximity operators in closed form. LTGL is extensively validated on synthetic data, achieving optimal performance in accuracy, structure learning and scalability with respect to ground truth and state-of-the-art methods for graphical inference. This chapter concludes with the application of LTGL to real case studies, from biology and finance, to illustrate how LTGL can be successfully employed to gain insights on multivariate time-series data.

Motivation

The problem of understanding complex systems arises in diverse contexts, such as financial markets (Liu, Han, and Zhang, 2012; Orchard, Agakov, and Storkey, 2013), social networks (Farasat et al., 2015) and biology (Hecker et al., 2009; Huang, Liao, and Wu, 2016; Lozano et al., 2009). In such contexts, the goal is to analyse the system in order to retrieve information on how the components behave. This requires accurate and interpretable mathematical models whose parameters, in practice, need to be estimated from observations.

Mathematically, as introduced in Chapter 2, a system can be modelled as a network of interactions (edges) between its entities (nodes). The underlying structure of the variables within the system is usually not known *a priori*. Nevertheless, observations of the system (*i.e.*, data) incorporate information on the interactions between variables, since they provide measurements of

such variables acting in the system.

The problem of inferring a network of variable interactions from data is known as *network inference* or *graphical model selection* (Friedman, Hastie, and Tibshirani, 2008; Lauritzen, 1996). During the last years the graphical modelling problem has received much attention, particularly for the availability of an always increasing number of samples that are required for a reliable network inference. Nonetheless, structure estimation of complex systems remains challenging for many reasons. This chapter tackles two particular aspects: *(i)* the presence of global hidden (or *latent*) factors, and *(ii)* the dynamics of systems that evolve over time. Arguably, the inference of a dynamical network encoding a complex system requires a specific attention to both aspects to result in a more realistic representation. In particular, a system may be affected by (latent) factors not encoded in the model. Such factors, acting in the system, influence how the observable entities behave and, hence, their inter-connections (Choi, Chandrasekaran, and Willsky, 2010). The consideration of hidden and unmeasured variables during the inference process emerges as crucial to avoid misrepresenting real-world data (Meng, Eriksson, and Hero, 2014).

At the same time, a complex system depends on a temporal component, which drives variable interactions to evolve consistently during its extent. This means that the structure can either change or remain stable according to the nature of the system itself. Hence, the understanding of a complex system is bound to the observation of its evolution. This is particularly evident in some applications, such as biology, where the interest could be to understand the response of the system to perturbation (Heyde et al., 2014; Molinelli et al., 2013).

Related Work

Latent variable models have been widely studied in literature, outperforming graphical models that only consider observable variables (Chandrasekaran, Parrilo, and Willsky, 2010; Choi et al., 2011; Yuan, 2012). At the same time, a set of methods were designed to study the temporal component through the inference of a dynamical network that incorporates prior knowledge on the behaviour of the system (Bianco-Martinez et al., 2016; Hallac et al., 2017; Sima, Hua, and Jung, 2009). Time-series with latent variables are considered to obtain a single graph which represents the global system (Anandkumar et al., 2013; Jalali and Sanghavi, 2011). However, to the best of the author's knowledge, state-of-the-art methods for regularised network inference do not consider simultaneously both time and latent variables during the inference of multiple connected networks.

Contribution

This chapter proposes the *latent variable time-varying graphical lasso* (LTGL), a model for dynamical network inference where the observable structure is influenced by latent factors. This can be seen as an attempt to generalise both

dynamical and latent variable network inference under a single unified framework. In particular, starting from a set of observations of a system at different time points, LTGL infers an interaction network of the observed variables under the influence of latent factors, while taking in consideration the temporal evolution of the system. The empirical interaction network is decomposed into the true underlying structure of the network and the contribution of latent factors, under the assumption that both observable variables and latent factors interdependence follow a temporal non-random behaviour. For this reason, the model allows to include prior knowledge on the evolutionary pattern of the system. The imposition of such prior knowledge benefits inference and subsequent analysis of the network, accentuating precise dynamical patterns. This is particularly important when the number of samples is low compared to the number of observed and latent variables in the system. In fact, the inference of the network at particular time points exploits the dependence between consecutive temporal states. Such advantage is achieved by a simultaneous inference of all the dynamical system, that, mathematically, translate into imposing constraints on the network behaviour. The following sections will provide a set of possible constraints that can be applied independently on both observed and latent components, allowing for a wide range of evolutionary patterns.

Figure 5.1 provides an example of the theoretical model assumed by LTGL. Here, observed and latent variables (x_i and z_i) are connected in a slightly different way at each time. Note that the observations of the system only involve variables x_i, while the hidden factors z_i influence the system without being actually observed. Hence, when analysing samples which are regulated from a dynamical network with hidden factors, it is infeasible to precisely infer the identity of latent variables, but only an estimation of their number and their effect on the global system can be obtained.

Starting from the theoretical model, the contribution of this chapter involves a minimisation algorithm based on ADMM (Boyd et al., 2010). The algorithm is divided into independent steps using proximal operators, which can be solved by closed-form solutions favouring a fast and scalable computation (Danaher, Wang, and Witten, 2014; Hallac et al., 2017; Ma, Xue, and Zou, 2013). Generic penalties for the problem at hand may be specified, with the only requirement that it is possible to express such constraints via proximal mappings.

The method is implemented in a Python framework, based on the use of highly optimised low-level libraries for numerical computations (Availability and Implementation). Experiments on synthetic data show LTGL to achieve optimal performance in relation to ground truth and to state-of-the-art methods for graphical modelling, in terms of accuracy, structure learning and scalability. Moreover, a particular emphasis will be put on the computational efficiency of LTGL while increasing the number of unknowns of the problem and the model complexity.

This chapter will conclude with the application of LTGL to real-world data sets to illustrate how LTGL can be successfully employed to gain insights

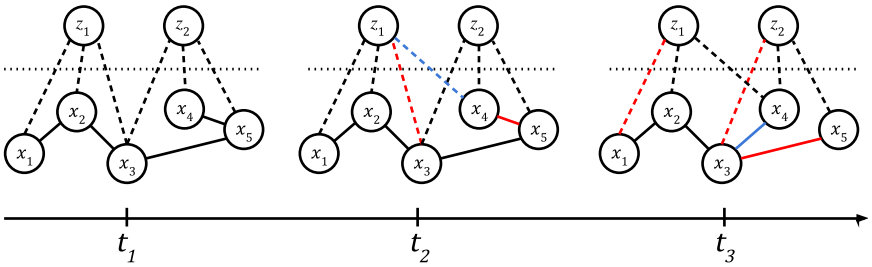


FIGURE 5.1. A dynamical network with latent factors z_i and observed variables x_i. At each time t_i, all connections between latent and observed variables (- - - - lines) and connections among observed variables (—— lines) may change according to a specific temporal behaviour. For simplicity, latent variables are here independent from each other (hence not connected). Blue/red colours indicate a new link is added to/removed from the network.

on multivariate time-series data. In particular, experiments involve biological and financial data sets, showing the use of LTGL in different contexts. In the first case, the analysis concerns *Escherichia coli* response to perturbation, correctly identified by LTGL. In the latter case, LTGL is used to investigate on a financial data set, showing how the contribution of latent factors is relevant to understand the behaviour of the system.

Outline

The rest of the chapter is organised as follows. Section 5.1 contains the theoretical formulation of the problem and the proposed method. Section 5.2 describes in details the optimisation algorithm for the minimisation of the functional. Sections 5.3 and 5.4 illustrate the use of LTGL on synthetic and real data, respectively. Section 5.5 concludes with a discussion on the results.

5.1 Model Formulation

This chapter proposes a novel statistical model for the inference of networks that change consistently in time under the influence of latent factors, called *latent variable time-varying graphical lasso* (LTGL). LTGL infers the dynamical network of complex systems by decomposing the problem into two parts — similarly to (Chandrasekaran, Parrilo, and Willsky, 2010) for static network inference. Two components of the dynamical network are considered: a true underlying structure on the observed variables and the contribution of latent factors. This allows to factor out the contribution of hidden variables, favouring a reliable modelling of the dynamical system. The novelty of such method is the simultaneous inference of a dynamical network with latent factors that exploits the imposition of behavioural consistency on both observed variables interactions and latent influence through the use of penalisation terms. This allows for an easier interpretation of the evolution of the dynamical system while, at the same time, improving its graphical modelling. The two separate

(while closely related) components at each time point are obtained by integrating the network inference with the information coming from temporally different states of the network.

Formally, let $X_t \in \mathbb{R}^{n_t \times d}$, for $t = 1, \ldots, T$, be a set of observations measured at T different time points composed by n_t samples of d observed variables. (Note that, for each time point t, samples are assumed to be drawn from the probability distribution on the observed variables conditioned on the latent ones.) Let $S_t = (1/n_t)X_t^\top X_t$ be the empirical covariance matrix at time t. The goal is to retrieve a set of sparse matrices $\boldsymbol{\Theta} = (\Theta_1, \ldots, \Theta_T)$ and a set of low-rank matrices $\boldsymbol{L} = (L_1, \ldots, L_T)$ such that, at each time point t, Θ_t encodes the conditional independences between the observed variables, while L_t provides the summary of marginalisation over latent variables on the observed ones (see Section 2.6).

Consider Equation (2.17) at a specific time t. Here, the goal is to impose continuity between the structure and the hidden variables contribution in time, so the difference between consecutive graphs is forced to abide certain constraints by adding two penalisation terms. The LTGL model takes the following form:

$$\begin{aligned} \underset{\substack{\Theta_t - L_t \in \mathcal{S}_{++}^d \\ L_t \in \mathcal{S}_+^d}}{\text{minimize}} \quad & \sum_{t=1}^{T} \Big[- n_t \, \ell(S_t, \Theta_t - L_t) + \alpha \|\Theta_t\|_{od,1} + \tau \|L_t\|_* \Big] \\ & + \beta \sum_{t=1}^{T-1} \Psi(\Theta_{t+1} - \Theta_t) + \eta \sum_{t=1}^{T-1} \Phi(L_{t+1} - L_t), \end{aligned} \tag{5.1}$$

where Ψ and Φ are penalty functions that force the structure of the network to change over time according to a certain behaviour by acting on Θ and L, respectively. Temporal consistency of both the structure of the network and latent factors contribution is guaranteed by the use of such penalty functions, which benefits the network inference in particular in presence of few available observations of the system.

Chapter 4 includes possible choices for Ψ and Φ. Their choice is arbitrary, based on prior knowledge on the evolution of the respective components in the system. Also, Ψ and Φ are independent, allowing LTGL to model a wide range of dynamical behaviours of complex systems.

5.2 Minimisation Method

Problem (5.1) is convex, provided that the penalty functions Ψ and Φ are convex, and it is coercive because of the regularisers. Thus, Problem (5.1) admits solutions. Nonetheless, its optimisation is challenging in practice due to the high number of unknown matrices involved ($2T$, for a total of $2T\frac{d(d+1)}{2}$ unknowns of the problem). A suitable method for the minimisation is ADMM (Boyd et al., 2010). It allows to decouple the variables obtaining a separable minimisation problem which can be efficiently solved in parallel. The sub-

problems exploit proximal operators which are (mostly) solvable in closed-form, leading to a simple iterative algorithm.

In order to decouple the involved matrices, let define three dual variables $\boldsymbol{R}$, $\boldsymbol{Z} = (\boldsymbol{Z}_1, \boldsymbol{Z}_2)$ and $\boldsymbol{W} = (\boldsymbol{W}_1, \boldsymbol{W}_2)$ and two projections:

$$\begin{aligned} P_1 \colon (\mathbb{R}^{d\times d})^T &\to (\mathbb{R}^{d\times d})^{T-1} & \qquad P_2 \colon (\mathbb{R}^{d\times d})^T &\to (\mathbb{R}^{d\times d})^{T-1} \\ \boldsymbol{A} &\mapsto (A_1, \dots, A_{T-1}) & \qquad \boldsymbol{A} &\mapsto (A_2, \dots, A_T) \end{aligned}$$

Problem (5.1) becomes:

$$\left.\begin{aligned} \underset{\substack{(\boldsymbol{\Theta},\boldsymbol{L},\boldsymbol{R},\boldsymbol{Z},\boldsymbol{W}) \\ \boldsymbol{L}_t \succeq 0}}{\text{minimize}} \quad & \sum_{t=1}^{T} \Big[- n_t \, \ell(S_t, R_t) + \alpha \left\|\Theta_t\right\|_{od,1} + \tau \left\|L_t\right\|_* \Big] \\ & + \beta \sum_{t=1}^{T-1} \Psi(Z_{2,t} - Z_{1,t}) + \eta \sum_{t=1}^{T-1} \Phi(W_{2,t} - W_{1,t}) \\ \text{s.t.} \quad & \boldsymbol{R} = \boldsymbol{\Theta} - \boldsymbol{L}, \;\; \boldsymbol{Z}_1 = P_1\boldsymbol{\Theta}, \;\; \boldsymbol{Z}_2 = P_2\boldsymbol{\Theta}, \;\; \boldsymbol{W}_1 = P_1\boldsymbol{L}, \;\; \boldsymbol{W}_2 = P_2\boldsymbol{L}. \end{aligned}\right\} \tag{5.2}$$

The corresponding augmented Lagrangian is as follows:

$$\begin{aligned} & \mathcal{L}_\rho(\boldsymbol{\Theta}, \boldsymbol{L}, \boldsymbol{R}, \boldsymbol{Z}, \boldsymbol{W}, \boldsymbol{U}) \\ & = \sum_{t=1}^{T} \Big[- n_t \, \ell(S_t, R_t) + \alpha \left\|\Theta_t\right\|_{od,1} + \tau \left\|L_t\right\|_* + \mathbb{I}(L \succeq 0) \Big] \\ & + \beta \sum_{t=1}^{T-1} \Psi(Z_{2,t} - Z_{1,t}) + \eta \sum_{t=1}^{T-1} \Phi(W_{2,t} - W_{1,t}) \\ & + \frac{\rho}{2} \sum_{t=1}^{T} \Big[\left\|R_t - \Theta_t + L_t + U_{0,t}\right\|_F^2 - \left\|U_{0,t}\right\|_F^2 \Big] \\ & + \frac{\rho}{2} \sum_{t=1}^{T-1} \Big[\left\|\Theta_t - Z_{1,t} + U_{1,t}\right\|_F^2 - \left\|U_{1,t}\right\|_F^2 + \left\|\Theta_{t+1} - Z_{2,t} + U_{2,t}\right\|_F^2 - \left\|U_{2,t}\right\|_F^2 \Big] \\ & + \frac{\rho}{2} \sum_{t=1}^{T-1} \Big[\left\|L_t - W_{1,t} + U_{3,t}\right\|_F^2 - \left\|U_{3,t}\right\|_F^2 + \left\|L_{t+1} - W_{2,t} + U_{4,t}\right\|_F^2 - \left\|U_{4,t}\right\|_F^2 \Big] \end{aligned} \tag{5.3}$$

where $\boldsymbol{U} = (\boldsymbol{U}_0, \boldsymbol{U}_1, \boldsymbol{U}_2, \boldsymbol{U}_3, \boldsymbol{U}_4)$ are the scaled dual variables. The ADMM algorithm for Problem (5.2) writes down as:

for $k = 1, \ldots$ **do**

$$\boldsymbol{R}^{k+1} = \arg\min_{\boldsymbol{R}} \ \mathcal{L}_\rho(\boldsymbol{\Theta}^k, \boldsymbol{L}^k, \boldsymbol{R}, \boldsymbol{Z}^k, \boldsymbol{W}^k, \boldsymbol{U}^k) \tag{5.4}$$

$$\boldsymbol{\Theta}^{k+1} = \arg\min_{\boldsymbol{\Theta}} \ \mathcal{L}_\rho(\boldsymbol{\Theta}, \boldsymbol{L}^k, \boldsymbol{R}^{k+1}, \boldsymbol{Z}^k, \boldsymbol{W}^k, \boldsymbol{U}^k) \tag{5.5}$$

$$\boldsymbol{L}^{k+1} = \arg\min_{\boldsymbol{L}} \ \mathcal{L}_\rho(\boldsymbol{\Theta}^{k+1}, \boldsymbol{L}, \boldsymbol{R}^{k+1}, \boldsymbol{Z}^k, \boldsymbol{W}^k, \boldsymbol{U}^k) \tag{5.6}$$

$$\boldsymbol{Z}^{k+1} = \begin{bmatrix} \boldsymbol{Z}_1^{k+1} \\ \boldsymbol{Z}_2^{k+1} \end{bmatrix} = \arg\min_{\boldsymbol{Z}} \ \mathcal{L}_\rho(\boldsymbol{\Theta}^{k+1}, \boldsymbol{L}^{k+1}, \boldsymbol{R}^{k+1}, \boldsymbol{Z}, \boldsymbol{W}^k, \boldsymbol{U}^k) \tag{5.7}$$

$$\boldsymbol{W}^{k+1} = \begin{bmatrix} \boldsymbol{W}_1^{k+1} \\ \boldsymbol{W}_2^{k+1} \end{bmatrix} = \arg\min_{\boldsymbol{W}} \ \mathcal{L}_\rho(\boldsymbol{\Theta}^{k+1}, \boldsymbol{L}^{k+1}, \boldsymbol{R}^{k+1}, \boldsymbol{Z}^{k+1}, \boldsymbol{W}, \boldsymbol{U}^k) \tag{5.8}$$

$$\boldsymbol{U}^{k+1} = \begin{bmatrix} \boldsymbol{U}_0^k \\ \boldsymbol{U}_1^k \\ \boldsymbol{U}_2^k \\ \boldsymbol{U}_3^k \\ \boldsymbol{U}_4^k \end{bmatrix} + \begin{bmatrix} \boldsymbol{R}^{k+1} - \boldsymbol{\Theta}^{k+1} + \boldsymbol{L}^{k+1} \\ P_1\boldsymbol{\Theta}^{k+1} - \boldsymbol{Z}_1^{k+1} \\ P_2\boldsymbol{\Theta}^{k+1} - \boldsymbol{Z}_2^{k+1} \\ P_1\boldsymbol{L}^{k+1} - \boldsymbol{W}_1^{k+1} \\ P_2\boldsymbol{L}^{k+1} - \boldsymbol{W}_2^{k+1} \end{bmatrix}. \tag{5.9}$$

5.2.1 R Step

The minimisation problem involving the matrix $\boldsymbol{R}$ of Equation (5.4) can be split into parallel updates, since $\mathcal{L}_\rho(\boldsymbol{\Theta}, \boldsymbol{L}, \boldsymbol{R}, \boldsymbol{Z}, \boldsymbol{W}, \boldsymbol{U})$ is separable in the variables $(R_1, \ldots, R_T)$. Hence, each R_t at iteration $k+1$ is given by:

$$\begin{aligned} R_t^{k+1} &= \arg\min_R \ \operatorname{tr}(S_t R) - \operatorname{logdet}(R) + \frac{\rho}{2n_t} \left\| R - \Theta^k + L^k + U_{0,t}^k \right\|_F^2 \\ &= \arg\min_R \ \operatorname{tr}(S_t R) - \operatorname{logdet}(R) + \frac{\rho}{2n_t} \left\| R - A_t^k \right\|_F^2 \\ &= \arg\min_R \ \operatorname{tr}(S_t R) - \operatorname{logdet}(R) + \frac{\rho}{2n_t} \left\| R - \frac{A_t^k + A_t^{k\top}}{2} \right\|_F^2 \end{aligned} \tag{5.10}$$

with $A_t^k = \Theta_t^k - L_t^k - U_{0,t}^k$. Note that the last equality in (5.10) follows from the symmetry of R — which also guarantees the logdet to be well-defined. Equation (5.10) can be explicitly solved. Indeed, Fermat's rule yields:

$$S_t - \frac{\rho}{n_t} \frac{A_t^k + A_t^{k\top}}{2} = R^{-1} - \frac{\rho}{n_t} R. \tag{5.11}$$

Then the solution to Equation (5.11) is (Danaher, Wang, and Witten, 2014; Hallac et al., 2017; Witten and Tibshirani, 2009):

$$R_t^{k+1} = \frac{n_t}{2\rho} V^k \left(-E^k + \sqrt{(E^k)^2 + \frac{4\rho}{n_t} I} \right) V^{k\top}$$

where $V^k E^k V^{k\top}$ is the eigenvalue decomposition of $S_t - \frac{\rho}{n_t}\frac{A_t^k + A_t^{k\top}}{2}$.

5.2.2 Θ Step

Likewise the $\boldsymbol{R}$ step, the update of $\boldsymbol{\Theta}$ in Equation (5.5) can be done in a parallel fashion, as follows:

$$\begin{aligned}\Theta_t^{k+1} &= \arg\min_{\Theta} \ \alpha\|\Theta\|_{od,1} + \frac{\rho}{2}\Bigg[\left\|R_t^k - \Theta + L_t^k + U_{0,t}^k\right\|_F^2 \\ &\qquad + \overline{\delta}_{tT}\left\|\Theta - Z_{1,t}^k + U_{1,t}^k\right\|_F^2 + \overline{\delta}_{t1}\left\|\Theta - Z_{2,t-1}^k + U_{2,t-1}^k\right\|_F^2\Bigg] \\ &= \arg\min_{\Theta} \ \alpha\|\Theta\|_{od,1} + (1 + \overline{\delta}_{tT} + \overline{\delta}_{t1})\frac{\rho}{2}\left\|\Theta - B_t^k\right\|_F^2\end{aligned} \tag{5.12}$$

where $\overline{\delta}_{ij} = 1 - \delta_{ij}$, with δ_{ij} Kronecker delta (that is equal to 1 when $i = j$, 0 otherwise) and

$$B_t^k = \frac{L_t^k + R_t^k + U_{0,t}^k + \overline{\delta}_{tT}(Z_{1,t}^k - U_{1,t}^k) + \overline{\delta}_{t1}(Z_{2,t-1}^k - U_{2,t-1}^k)}{1 + \overline{\delta}_{tT} + \overline{\delta}_{t1}}.$$

Problem (5.12) is solved as:

$$\Theta_t^{k+1} = \text{prox}_{\zeta\|\cdot\|_{od,1}}(B_t^k) = S_\zeta(B_t^k)$$

with $\zeta = \frac{\alpha}{\rho(1+\overline{\delta}_{tT}+\overline{\delta}_{t1})}$, and $S_\zeta(\cdot)$ element-wise off-diagonal soft-thresholding function.

5.2.3 L Step

The parallel update of $\boldsymbol{L}$ in Equation (5.6) can be written as:

$$\begin{aligned}L_t^{k+1} &= \arg\min_{L} \ \tau\,\text{tr}(L) + \mathbb{I}(L \succeq 0) + \frac{\rho}{2}\Bigg[\left\|R_t^{k+1} - \Theta_t^{k+1} + L + U_{0,t}^k\right\|_F^2 \\ &\qquad + \overline{\delta}_{tT}\left\|L - W_{1,t}^k + U_{3,t}^k\right\|_F^2 + \overline{\delta}_{t1}\left\|L - W_{2,t-1}^k + U_{4,t-1}^k\right\|_F^2\Bigg] \\ &= \arg\min_{L} \ \tau\,\text{tr}(L) + \mathbb{I}(L \succeq 0) + (1 + \overline{\delta}_{tT} + \overline{\delta}_{t1})\frac{\rho}{2}\left\|L - C_t^k\right\|_F^2 \\ &= \arg\min_{L} \ \tau\,\text{tr}(L) + \mathbb{I}(L \succeq 0) + (1 + \overline{\delta}_{tT} + \overline{\delta}_{t1})\frac{\rho}{2}\left\|L - \frac{C_t^k + C_t^{k\top}}{2}\right\|_F^2\end{aligned} \tag{5.13}$$

where

$$C_t^k = \frac{\Theta_t^{k+1} - R_t^{k+1} - U_{0,t}^k + \overline{\delta}_{tT}(W_{1,t} - U_{3,t}) + \overline{\delta}_{t1}(W_{2,t-1} - U_{4,t-1})}{1 + \overline{\delta}_{tT} + \overline{\delta}_{t1}}.$$

Note that the last equality in (5.13) follows from the symmetry of L. The solution to Problem (5.13) is (Ma, Xue, and Zou, 2013):

$$L_t^{k+1} = V^k \tilde{E} V^{k\top},$$

where $V^k E^k V^{k\top}$ is the eigenvalue decomposition of C_t^k, and

$$\tilde{E}_{jj} = \max\left(E_{jj}^k - \frac{\tau}{\rho(1 + \overline{\delta}_{tT} + \overline{\delta}_{t1})},\ 0\right).$$

5.2.4 Z and W Step

The dual variables $\boldsymbol{Z}$ and $\boldsymbol{W}$ enforce the network to behave in time consistently with the choice of Ψ and Φ, respectively. $\boldsymbol{Z}$ is the dual variable of $\boldsymbol{\Theta}$ while $\boldsymbol{W}$ is the dual variable of $\boldsymbol{L}$. The update of $\boldsymbol{Z}$ and $\boldsymbol{W}$ are similar, so this section will cover both of them.

The dual variable $\boldsymbol{Z}$ is defined as $(\boldsymbol{Z}_1, \boldsymbol{Z}_2)$. Such matrices are not separable in Equation (5.2), thus they must be jointly updated. The update of $\boldsymbol{Z}$ in Equation (5.7) can be rewritten as:

$$\begin{aligned}\begin{bmatrix} Z_{1,t}^{k+1} \\ Z_{2,t}^{k+1} \end{bmatrix} = \underset{Z_1, Z_2}{\arg\min}\ \ & \beta\, \Psi(Z_2 - Z_1) + \frac{\rho}{2}\left\|\Theta_t^k - Z_1 + U_{1,t}^k\right\|_F^2 \\ & + \frac{\rho}{2}\left\|\Theta_{t+1}^k - Z_2 + U_{2,t}^k\right\|_F^2.\end{aligned} \tag{5.14}$$

Let $\hat{\Psi}\begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} = \Psi(Z_2 - Z_1)$. Then, Problem (5.14) can be solved with an unique update (Hallac et al., 2017):

$$\begin{bmatrix} Z_{1,t}^{k+1} \\ Z_{2,t}^{k+1} \end{bmatrix} = \text{prox}_{\frac{\beta}{\rho}\hat{\Psi}(\cdot)}\left(\begin{bmatrix} \Theta_t^k + U_{1,t}^k \\ \Theta_{t+1}^k + U_{2,t}^k \end{bmatrix}\right).$$

The same holds for the $\boldsymbol{W}$ step. The update of $\boldsymbol{W}$ becomes:

$$\begin{aligned}\begin{bmatrix} W_{1,t}^{k+1} \\ W_{2,t}^{k+1} \end{bmatrix} = \underset{W_1, W_2}{\arg\min}\ \ & \eta\, \Phi(W_2 - W_1) + \frac{\rho}{2}\left\|L_t^k - W_1 + U_{3,t}^k\right\|_F^2 \\ & + \frac{\rho}{2}\left\|L_{t+1}^k - W_2 + U_{4,t}^k\right\|_F^2.\end{aligned} \tag{5.15}$$

Let $\hat{\Phi}\begin{bmatrix} W_1 \\ W_2 \end{bmatrix} = \Phi(W_2 - W_1)$. Hence, the proximal operator for the update of $W_{1,t}$ and $W_{2,t}$ becomes:

$$\begin{bmatrix} W_{1,t}^{k+1} \\ W_{2,t}^{k+1} \end{bmatrix} = \text{prox}_{\frac{\eta}{\rho}\hat{\Phi}(\cdot)}\left(\begin{bmatrix} L_t^k + U_{3,t}^k \\ L_{t+1}^k + U_{4,t}^k \end{bmatrix}\right).$$

For the particular derivation of different proximal operators, see (Hallac et al., 2017).

5.2.5 Termination Criterion

According to (Boyd et al., 2010), the algorithm is said to converge if the primal and dual residuals are sufficiently small, *i.e.*, if $\|r^k\|_2^2 \leq \epsilon^{\text{pri}}$ and $\|s^k\|_2^2 \leq \epsilon^{\text{dual}}$. At each iteration $k > 2$ these values are computed as follows:

$$\begin{aligned}
\|r^k\|_2^2 &= \|\boldsymbol{R}^k - \boldsymbol{\Theta}^k + \boldsymbol{L}^k\|_F^2 + \|P_1\boldsymbol{\Theta}^k - \boldsymbol{Z}_1^k\|_F^2 + \|P_2\boldsymbol{\Theta}^k - \boldsymbol{Z}_2^k\|_F^2 \\
&\quad + \|P_1\boldsymbol{L}^k - \boldsymbol{W}_1^k\|_F^2 + \|P_2\boldsymbol{L}^k - \boldsymbol{W}_2^k\|_F^2 \\
\|s^k\|_2^2 &= \rho\big(\|\boldsymbol{R}^k - \boldsymbol{R}^{k-1}\|_F^2 + \|\boldsymbol{Z}_1^k - \boldsymbol{Z}_1^{k-1}\|_F^2 + \|\boldsymbol{Z}_2^k - \boldsymbol{Z}_2^{k-1}\|_F^2 \\
&\quad + \|\boldsymbol{W}_1^k - \boldsymbol{W}_1^{k-1}\|_F^2 + \|\boldsymbol{W}_2^k - \boldsymbol{W}_2^{k-1}\|_F^2\big) \\
\epsilon^{\text{pri}} &= c + \epsilon^{\text{rel}} \max\big(D_1, D_2\big) \\
\epsilon^{\text{dual}} &= c + \epsilon^{\text{rel}} \rho\big(D_3^k\big)
\end{aligned}$$

where $c = \epsilon^{\text{abs}} d(5T-4)^{1/2}$, ϵ^{abs} and ϵ^{rel} are arbitrary tolerance parameters, and

$$\begin{aligned}
\left\|D_1^k\right\|_F^2 &= \|\boldsymbol{R}^k\|_F^2 + \|\boldsymbol{Z}_1^k\|_F^2 + \|\boldsymbol{Z}_2^k\|_F^2 + \|\boldsymbol{W}_1^k\|_F^2 + \|\boldsymbol{W}_2^k\|_F^2, \\
\left\|D_2^k\right\|_F^2 &= \|\boldsymbol{\Theta}^k - \boldsymbol{L}^k\|_F^2 + \|P_2\boldsymbol{\Theta}^k\|_F^2 + \|P_1\boldsymbol{\Theta}^k\|_F^2 + \|P_2\boldsymbol{L}^k\|_F^2 + \|P_1\boldsymbol{L}^k\|_F^2, \\
\left\|D_3^k\right\|_F^2 &= \|\boldsymbol{U}_0^k\|_F^2 + \|\boldsymbol{U}_1^k\|_F^2 + \|\boldsymbol{U}_2^k\|_F^2 + \|\boldsymbol{U}_3^k\|_F^2 + \|\boldsymbol{U}_4^k\|_F^2.
\end{aligned}$$

5.2.6 Varying ρ

The parameter ρ can be updated at each iteration with the following schema:

$$\rho^{k+1} = \begin{cases} \tau^{\text{incr}}\rho^k & \text{if } \|r^k\|_2 > \mu\|s^k\|_2 \\ \rho^k/\tau^{\text{decr}} & \text{if } \|s^k\|_2 > \mu\|r^k\|_2 \\ \rho^k & \text{otherwise,} \end{cases}$$

where $\mu > 1$, $\tau^{\text{incr}} > 1$, $\tau^{\text{decr}} > 1$ are parameters of the algorithm, specified in advance. This has been shown to improve convergence in practice (Boyd et al., 2010). Also, when employing a varying penalty ρ with the scaled ADMM form, the scaled dual variables $\boldsymbol{U}$ are rescaled after updating ρ.

5.3 *Experiments*

Extensive experiments on synthetic data assessed the performance of the method in terms of structure recovery and measure of latent variables influence. The performance of LTGL was evaluated with respect to the ground truth and to state-of-the-art methods for graphical inference. In particular, two particular aspects of LTGL were assessed, that are *modelling performance* and *scalability*, in separated experiments. Modelling performance was estimated by comparing the inferred graphical model to the true network underlying the data set. The scalability experiment, instead, assessed the computational time for convergence needed for increasing problem complexity.

5.3.1 Modelling Performance

The modelling performance of LTGL was evaluated on two synthetic data sets. The ground truth sets of matrices $\boldsymbol{\Theta} = (\Theta_1, \ldots, \Theta_T)$ and $\boldsymbol{L} = (L_1, \ldots, L_T)$ were obtained by perturbing initial matrices Θ_1 and L_1, according to a specific behaviour for $T - 1$ times, guaranteeing that $\Theta_t - L_t \succ 0$ and $L_t \succeq 0$ for $t = 1, \ldots, T$. The initial matrices were generated according to (Yuan, 2012), following the form (2.16). Θ_1 and L_1 correspond to Θ_O and $\Theta_{OH}\Theta_H^{-1}\Theta_{HO}$, respectively, with Θ_H identity matrix and $\Theta_{HO} = \Theta_{OH}^\top$. Note that, since Θ_H has full rank, the number of latent variables is H. In particular, for d observed variables, n samples and T timestamps, I generated a data set $X \in (\mathbb{R}^{n\times d})^T$ sampled from T multivariate normal distributions $X_t \sim \mathcal{N}_t(\mathbf{0}, \Sigma_t)$, for $i = 1, \ldots, T$, and $\Sigma_t^{-1} = \Theta_t - L_t$.

5.3.1.1 ℓ_2^2 Perturbation (p_2)

The first data set was generated by perturbing the initial matrices with a random matrix of small ℓ_2^2 norm. This perturbation assumes the differences between two consecutive matrices to be small and bounded over time, *i.e.*, $\|\Theta_t - \Theta_{t-1}\|_F \leq \epsilon$ for $i = 2, \ldots, T$. The bound ϵ on the norm is chosen *a priori*.

The update of L_t is done maintaining consistency with the theoretical model where $L_t = \Theta_{OH,t}\Theta_{OH,t}^\top$. Indeed, the update adds a random matrix with a small norm to $\Theta_{OH,t-1}$. In this way, the rank of L_t remains the same as the number of latent variables and constant over time. Data were generated in $\mathbb{R}^{100}$ with 10 time stamps, conditioned on 20 latent variables. For each time stamp, 100 samples were drawn from the distribution. For this reason, in this setting, the contribution of latent factors is predominant with respect to the network evolution in time.

5.3.1.2 ℓ_1 Perturbation (p_1)

A second data set was generated according to a different perturbation model. Here, the precision matrix was updated by randomly choosing an edge and swapping its state, *i.e.*, by removing or adding a connection between two variables. This allows for a ℓ_1-norm evolutionary pattern of the network. Data were generated in $\mathbb{R}^{50}$ with 100 time stamps, conditioned on 5 latent variables. For each time stamp, 100 samples were drawn from the distribution. In this setting, the time consistency affects the network more than the latent factor contribution.

5.3.1.3 Scores

LTGL was evaluated using different scores measuring the divergence of the results from the ground truth. In particular, the performance was evaluated in terms of F_1 score, accuracy, mean rank error and mean squared error. Let *true/false positive* be the number of correctly/incorrectly existing inferred

TABLE 5.1. Performance in terms of F_1 score, accuracy (ACC), mean rank error (MRE) and mean squared error (MSE) of LTGL with respect to TVGL, LVGLASSO and GL. LTGL and TVGL are employed with both ℓ_2^2 and ℓ_1 penalties, to show how the prior on the evolution of the network affects the outcome.

perturbation	method	score			
		F_1	ACC	MRE	MSE
ℓ_2^2 (p_2)	LTGL (ℓ_2^2)	**0.926**	**0.994**	**0.70**	0.007
	LTGL (ℓ_1)	0.898	0.993	**0.70**	0.007
	TVGL (ℓ_2^2)	0.791	0.980	-	**0.003**
	TVGL (ℓ_1)	0.791	0.980	-	**0.003**
	LVGLASSO	0.815	0.988	2.80	0.007
	GL	0.745	0.974	-	0.004
ℓ_1 (p_1)	LTGL (ℓ_2^2)	0.842	0.974	0.29	0.013
	LTGL (ℓ_1)	**0.880**	**0.981**	**0.28**	0.013
	TVGL (ℓ_2^2)	0.742	0.950	-	0.009
	TVGL (ℓ_1)	0.817	0.968	-	0.009
	LVGLASSO	0.752	0.964	0.74	0.013
	GL	0.748	0.951	-	**0.007**

edges, *true/false negative* the number of correctly/incorrectly missing inferred edges (Hecker et al., 2009). The scores were computed as follows:

- F_1 score: indicates the quality of structure inference, as the harmonic mean of precision and recall.
- Accuracy (ACC): evaluates the number of true existing and missing connections in the network correctly inferred with respect to the total number of connections.
- Mean rank error (MRE): estimates the precision on the number of inferred latent variables, based on the rank of the set of matrices $\tilde{\boldsymbol{L}}$ in relation to the ground truth. The MRE score is defined as:

$$\text{MRE} = \frac{1}{T}\sum_{t=1}^{T}\left|\text{rank}(L_t) - \text{rank}(\tilde{L}_t)\right|.$$

  A value close to 0 means that LTGL is inferring the true number of latent variables over time, while, viceversa, a high value indicates a poor consideration of the contribution of the latent variables.
- Mean squared error (MSE): scores how close is the inferred precision matrix $\tilde{\boldsymbol{\Theta}}$ to the ground truth, in terms of the Frobenius norm:

$$\text{MSE} = \frac{2}{Td(d-1)}\sum_{t=1}^{T}\left\|\Theta_t^{(u)} - \tilde{\Theta}_t^{(u)}\right\|_2^2,$$

  where $\Theta^{(u)}$ denotes the upper triangular part of Θ.

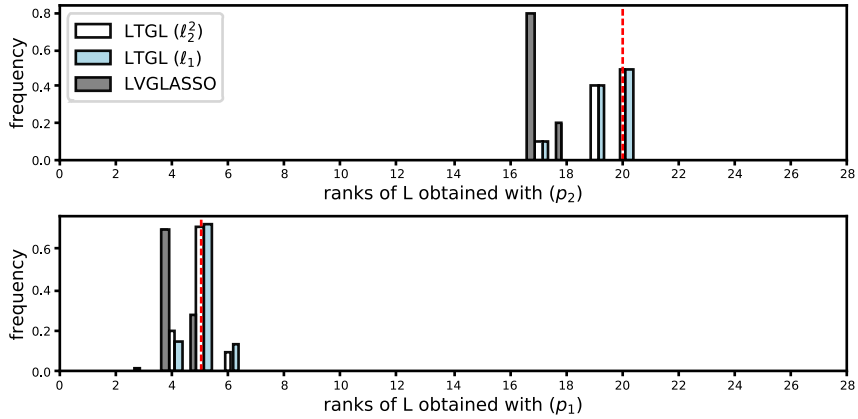


FIGURE 5.2. Distribution of inferred ranks over time. For each method that considers latent variables, I report the frequency of finding a specific rank during the network inference. The vertical line indicates the ground truth rank, around which all detected ranks lie. Note that, in (p_2), $L_t \in \mathbb{R}^{100\times100}$, so the range of possible ranks is $[0, 100]$. For (p_1), $L_t \in \mathbb{R}^{50\times50}$, hence the range is $[0, 50]$.

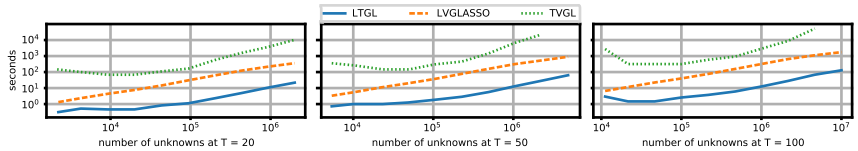


FIGURE 5.3. Scalability comparison for LTGL in relation to other ADMM-based methods. The compared methods are initialised in the same manner, *i.e.*, with all variable interactions (not self-interacting) set to zero. The computational time required for hyper-parameters selection is ignored. For LVGLASSO and TVGL, their relative original implementations were used. LTGL outperforms the other methods for each increasing time and dimensionality of the problem.

5.3.1.4 Discussion

Table 5.1 shows the performance of LTGL compared to graphical lasso (GL) (Friedman, Hastie, and Tibshirani, 2008), latent variable graphical lasso (LVGLASSO) (Chandrasekaran, Parrilo, and Willsky, 2010; Ma, Xue, and Zou, 2013) and time-varying graphical lasso (TVGL) (Hallac et al., 2017) in terms of F_1 score, accuracy, mean rank error (MRE) and mean squared error (MSE), for both settings with ℓ_2^2 (p_2) and ℓ_1 (p_1) perturbation. Note that MRE is not available for all the methods since neither GL nor TVGL consider latent factors. LTGL and TVGL are used with two temporal penalties according to the different perturbation models of data generation. This shows how the correct choice of the penalty for the problem at hand results in a more accurate network estimation. In both (p_2) and (p_1), LTGL outperforms the other methods for graphical modelling. In (p_2), in particular, LTGL correctly infers almost 99, 5% of edges in all the dynamical network both with the ℓ_2^2 and ℓ_1 penal-

ties. Nonetheless, the use of ℓ_2^2 penalty enhance the quality of the inference as expected from the theoretical assumption made during data generation. The choice of a proper penalty for the problem and the consideration of time consistency is reflected also in a low MRE, which encompasses LVGLASSO ability in detecting latent factors (Figure 5.2). In (p_2), in fact, the number of latent variables with respect to both observed variables and samples is high. Therefore, by exploiting temporal consistency of the network, LTGL is able to improve the latent factors estimation. Simultaneous consideration of time and latent variable also positively influences the F_1 score, *i.e.*, structure detection.

Above considerations also hold for the (p_1) setting. Here, LTGL achieves the best results in both F_1 score and accuracy, while having a low MRE. The adoption of ℓ_1 penalty improves structure estimation and latent factors detection, consistently with the data generation model. Such settings were designed to show how the prevalence of latent factors contribution or time consistency affects the outcome of a network inference method. In (p_2), where the latent factors contribution is prevalent, network inference is more precise when considering latent factors. In (p_1), instead, the number of time points is more relevant than the contribution of latent factors, hence it is more effective to exploit time consistency (both for latent and observed variables), evident from the results of Table 5.1. LTGL benefits from both aspects, leading to a noticeable improvement of the graphical modelling.

5.3.2 Scalability

This section shows a scalability analysis using LTGL with respect to different ADMM-based solvers. I evaluated the performance of LTGL in relation to LVGLASSO and TVGL, both implemented with closed-form solutions to ADMM subproblems. In general, the complexity of the three compared solvers is the same (up to a constant). The implementation of GL was not included in such experiment, since it is not based on ADMM but on coordinate descent hence not comparable to LTGL. As in Section 5.3.1, I generated different data sets $X \in (\mathbb{R}^{n \times d})^T$ with different values of T and d. In particular, $d \in [10, 400)$ and $T = \{20, 50, 100\}$. Experiments ran on a machine provided with two CPUs (2.4 GHz, 8 cores each).

Figure 5.3 shows, for the three different time settings, the scalability of the methods in terms of seconds per convergence considering different number of unknowns of the problem (*i.e.*, $2T\frac{d(d+1)}{2}$ with d observed variables and T times). In all settings, LTGL outperforms LVGLASSO and TVGL in terms of seconds per convergence. In particular, the computational time for convergence remains stable disregarding the number of time points under consideration. I emphasise that the most computationally expensive task performed by LTGL solver consists in two eigenvalue decompositions, with a complexity of $O(d^3)$, to solve both $\boldsymbol{R}$ and $\boldsymbol{L}$ steps (Section 5.2).

5.3.3 Model Selection

The hyper-parameters of the method have been selected by using a Monte Carlo cross-validation (MCCV) procedure (Section 3.2.1). For each hyper-parameter combination, the model was trained on the learning set and the likelihood of the model was estimated on the independent test set. Formally, the score is defined as follows:

$$\text{score} = \sum_{t=1}^{T} \text{log-likelihood}(S_t^{\text{ts}}, \Theta_t - L_t). \tag{5.16}$$

The choice of hyper-parameters is based on the average maximum likelihood of the model across multiple splits of the data set.

However, the number of possible combinations of LTGL hyper-parameters can be arbitrarily large. In order to avoid the assessment of a grid of models (which can be computationally expensive), the choice of the best combination of hyper-parameters for each analysed data set relies on a Gaussian process-based Bayesian optimisation procedure, based on the expected improvement strategy (see Section 3.3.3) (Snoek, Larochelle, and Adams, 2012). In practice, assuming the dynamics of a real system to be unknown, it is possible to select the most appropriate temporal penalty by exploiting the same principles, *i.e.*, via a model selection procedure based on the likelihood of different temporal models.

5.4 *Applications to Real Data*

I applied LTGL to two real data sets, to show how the method can be employed to infer useful insights on multivariate time-series data. These data sets measure complex dynamical systems of different (biological and financial) nature, which are usually highly dimensional and feature complicated interdependences between variables. This fact makes them ideal candidates for an analysis using graphical models.

5.4.1 Metabolomic Data

The physiology of *Escherichia coli* necessitates rapid changes of its cellular and molecular network to adapt to environmental conditions. *E. coli* is widely studied because of the efficiency in its system response to perturbation. Following the analysis of (Jozefczuk et al., 2010), I used LTGL on *E. coli* data to infer network modifications across different time points evaluated before and after the application of environmental condition perturbations. I analysed the behaviour of metabolites, which have been shown to change consistently after the perturbation. Samples underwent one of two types of stress, namely cold and heat stress.

5.4.1.1 Perturbation Response Detection

I inferred the dynamical network of *E. coli* metabolites using LTGL with a group lasso (ℓ_2) penalty on latent variable contribution and a Laplacian (ℓ_2^2) penalty on the observed network. In this way, latent variables (which, in the model, could represent the stress or other factors) are allowed to change their global influence at a specific time point, while remaining stable in all others. At the same time, by conditioning the network on the latent variables, the observed network structure is allowed to change smoothly in time. Hence, we expect to see a global shift of the network between the second and third time points, that is when the perturbation has been introduced in the system. Figure 5.4 (a) shows the temporal deviation between time points, both for Θ, L and the total observed system $R = \Theta - L$. Latent variables temporal deviation reaches a peak at time t_{2-3}, right after the application of the perturbation to the system. Instead, the difference between consecutive Θs remains more stable. Consistently, the difference between the observed networks R shows a major change at the same time point. Hence we can distinguish the underlying evolving structure of metabolites while detecting the contribution of the latent variables which affect mostly the total system. In accordance with (Jozefczuk et al., 2010), the result shows a interaction between isoleucine, threonine, phenylalanine and 2-aminobutyric acid during the adaptation phase following the stress response (Figure 5.4, b). Hence, I can conclude that LTGL successfully inferred a dynamical network which adjusts in response to perturbation, in accordance with the prior knowledge about *E. coli* behaviour.

5.4.2 Stock Market

Finance is another example of a complex dynamical system suitable to be analysed through a graphical model. Stock prices, in particular, are highly related to each other and subject to time and environmental changes, *i.e.*, events that modify the system behaviour but are not directly related to companies share values (Bai and Ng, 2006). Such environmental changes could be seen as latent variables that act on the system without being actually part of it. Here, the assumption is that each company, while being part of a global financial system, is directly dependent from only a subset of others. For example, one can reasonably expect that stock prices of a technology company are not directly influenced by trend of companies on the primary sector. The modelling power of LTGL allows to detect both the evolution of relations between companies and environmental changes happening at a particular time point. In order to show this, I analysed stock prices[1] during the financial crisis of 2007–2008. The experiment was designed to consider the latent influence of the market drop on technology companies interactions.

[1]Data are freely available on https://quantquote.com/historical-stock-data.

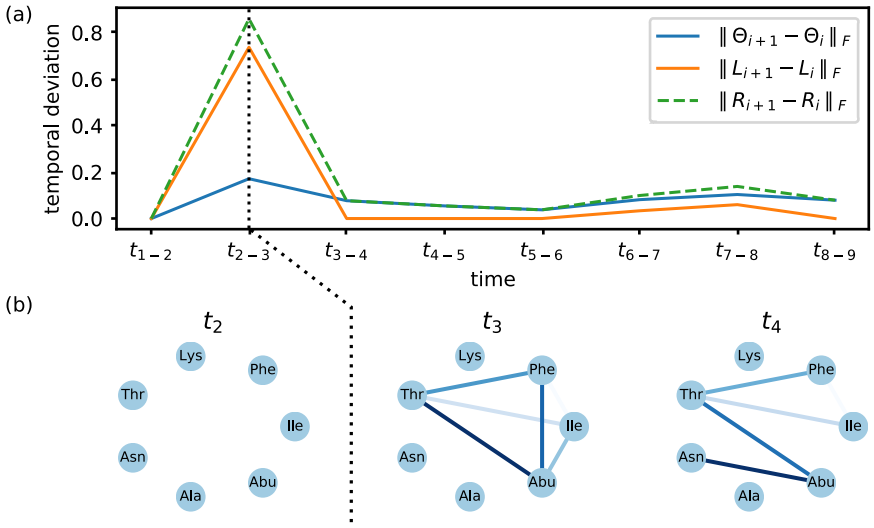


FIGURE 5.4. Structure change of *E. coli* metabolites subject to stress. The perturbation happens between time $t = 2$ and $t = 3$ (vertical dotted line). (a) Temporal deviation where each point represents the difference between the network at subsequent time points. The highest deviation on the observed network R appears when the stress was applied. This can be decomposed into two parts, the latent factors L and the underlying structure of observed variables Θ. (b) Structural changes of metabolites interactions before and after the perturbation.

5.4.2.1 Global Market Crisis Detection

I used a group lasso (ℓ_2) penalty to detect global shifts of the network. Figure 5.5 shows two major changes in both components of the network (latent and observed), in correspondence of late 2007 and late 2008. In particular, during October 2008 a global crisis of the market occurred, and this effect is especially evident for the shift of latent variables. Also, the observed network changes in correspondence of the latent variables shift or immediately after, caused by the effect of the crisis on the stock market. The latent factors influence explains how the change of the network was due to external factors that globally affected the market, and not to normal evolution of companies relationships. I further investigated on the causes for the first shift. Indeed, I found that in late 2007 it happened a drop of a big American company that was later pointed out as the beginning of the global crisis of the following year.

5.5 *Discussion*

This chapter proposes a novel method for graphical modelling of multivariate time-series. The model considers simultaneously the contribution of latent factors and time consistency in evolving systems. Indeed, such work is an attempt to generalise both latent variable and dynamical network inference. To this aim, the model imposes prior knowledge on the problem through penalty terms that force precision and latent matrices to be consistent in

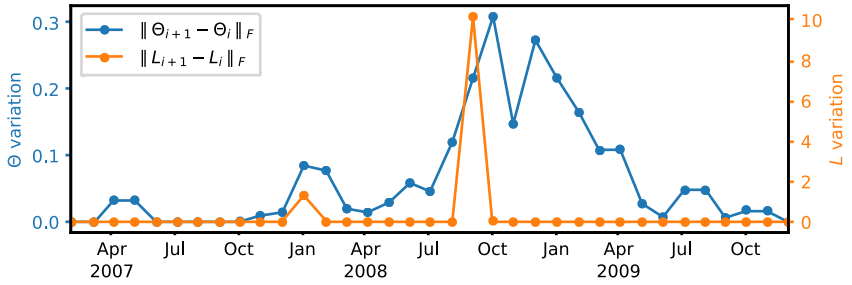


FIGURE 5.5. Temporal deviation for stock market data. Two peaks are observable, in correspondence of late 2007 and late 2008, when the financial crisis happened.

time. The choice of proper penalty terms maintains the convexity of the minimised functional and, along with the coercivity given by the regularisers, it guarantees global convergence of the proposed minimisation algorithm. Extensive experiments illustrate the ability of LTGL for the graphical modelling of synthetic and real-world data, where the possibility to decompose the total network into two separated components allows for a better understanding of the underlying phenomenon.

From a computational point of view, the choice of ADMM allows for a separable algorithm, where its steps have closed-form solutions. ADMM is an optimisation algorithm which can be easily implemented to run in parallel or even in a distributed system. At the moment, the implementation relies only on single-machine parallelisation, but further extensions may involve distribution on multiple computing machines. Nonetheless, the existing implementation is fast and scalable, as based on highly optimised libraries for numerical computation.

The proposed framework is modular in the choice of the penalties, allowing for precise modelling of different and complex behaviours of the system. This allows for a straightforward inclusion of additional penalty terms, based on the prior knowledge on the problem at hand. Possible extensions would involve alternative evolutionary models for different complex systems, *e.g.*, forcing subgroups of variables to behave consistently in time (Bolstad, Van Veen, and Nowak, 2011).

These could lead to interesting results in time-series clustering and pattern discovery. Such developments may increase the expression power of the method, leading to advances in data mining and to potential applications in diverse science fields.

Part III

Applications

Recent advances on time-series graphical modelling can be used in real contexts, in particular for biomedical data, which offers a challenging framework for the graphical inference methods. This part includes the work developed by exploiting temporal graphical models as described in Chapters 4 and 5. In particular, such methods were applied on breast invasive carcinoma RNA-seq data set (Chapter 6) and on haematopoietic stem cells (Chapter 7). Part of the work of Chapter 6 is included in (Tomasi, Squillario, and Barla, 2017).
Chapter 8 shows an application of Wishart processes for epilepsy data. Part of such chapter, in particular Section 8.2, is included in (D'Amario et al., 2018).

6 *Breast Cancer Evolution*

Understanding molecular variables which discriminate a set of clinical outcomes is not sufficient to exhaustively explain the molecular mechanisms that lead to different biological conditions. In fact, the interplay and interaction of molecular variables plays a key role in characterising a clinical outcome.

Usually, a set of variables which are able to characterise the biological conditions under analysis is identified, and then enrichment analysis is used *a posteriori* for a functional assessment of selected variables. In this case, prior knowledge on the interplay of such variables (*i.e.*, involvement in a common pathway) validates the result (by means of functional characterisation) instead of leading the analysis from the start.

This approach has some drawbacks. Misguided variable selection and classification procedures may include false hits, or, even worse, exclude variables potentially relevant in the biological context under analysis. In particular, this may happen when variables are not important *per se*, but may be deemed biologically relevant if considered within a molecular module. Also, when the number of samples is low compared to the number of variables the use of prior knowledge on the problem plays a fundamental role to direct the analysis and obtain a reliable model for the data at hand.

Motivation

Prior biological knowledge can be effectively exploited to learn the statistical model underlying the data, guaranteeing the non-exclusion of variables that are biologically relevant to the analysed disease (Zycinski et al., 2013). Such procedure may improve the variable selection and classification phase, by assigning importance to variables given prior biological knowledge.

A situation where the use of prior information has a key role for a reliable inference of a statistical model is in the context of graphical models. Graphical modelling of time-series data allow to understand how interactions between variables evolve over time.

In many real cases data do not correspond directly to time-series, meaning that the number of samples does not derive from a consecutive sampling of the same measured variables. However, samples can be considered in an appropriate ordering based on pseudotime, that is an arbitrary ordering to model, for example, an evolutionary progression.

Consider biological data, for which samples are associated to the information on the stage of a particular disease. In this context, a reasonable assumption is that samples belonging to a certain stage of progression of the disease may develop subsequent stages of the same disease. Hence, the idea is to use algorithms designed for time-series data also with samples following a

pseudotemporal ordering, where assumptions on the data distribution at each time-step still hold.

Contribution

This chapter comprises two data analysis pipelines for the identification of meaningful signalling pathways, reconstruction and quantitative assessment of the networks corresponding to the molecular variables in different tumour progression stages. Both pipelines exploit explicit prior knowledge on the problem, considering the information coming from signalling pathways during the learning phase.

After, groups of variables relevant for the problem are analysed using both pairwise network inference and graphical modelling methods, namely ARACNE (Margolin et al., 2006) and the latent variable time-varying graphical lasso (Chapter 5), to assess their evolving interactions during the different biological conditions.

A main difference between the pipelines regards the consideration of the different stages of the disease. The first pipeline, detailed in Section 6.2, regards the stages as different classes of the disease, interpreting the learning task as a multiclass classification. For each pathway specified a priori, a logistic regression model estimates the relevance of the pathway for the learning task. Then, the pipeline infers a list of the most informative pathways, before using such pathways for the next network inference task. A limitation of such pipeline, however, is that pathways (as inferred by the resampling strategy) are difficult to assess, given the dependency of performance on the number of repetitions of the experiment and on the metric employed.

The second learning pipeline (Sections 6.3.1 and 6.3.2) makes explicit use of the temporal ordering which allows to use the graphical models for time-series analysis developed in Chapter 5. Also, instead of relying on different learning tasks for each pathways (Section 6.2), this pipelines interprets the pathway selection as a regularisation task, exploiting the ℓ_1-norm to restrict the model to include only relevant pathways able to discriminate between the classes of samples, in such a way to select relevant pathways for the learning task at once.

Outline

The rest of the chapter is organised as follows. Section 6.1 introduces the data considered in the analysis. Section 6.2 describes the first learning pipeline, based on a multiclass learning problem for pathway selection and pairwise network inference. Section 6.3 describes the second pipeline, composed of a pathway selection step using the group lasso with overlap (Section 6.3.1) and a graphical modelling step used for a classification task (Section 6.3.2), and the results obtained on the breast invasive carcinoma data set (Section 6.3.3). Section 6.4 concludes with a summary of the pipelines described in this chapter.

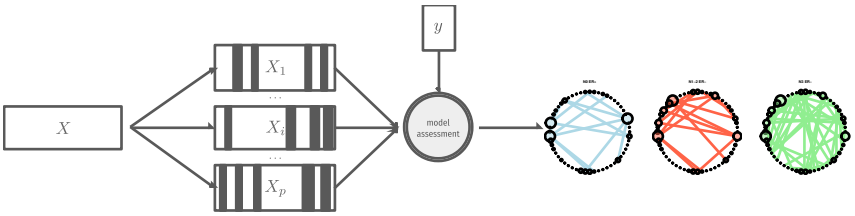


FIGURE 6.1. Knowledge driven graphical inference pipeline. The data set X is divided based on the p pathways. After the model assessment step, the most performing pathways are selected to generate a network of interactions that model the system for each class of samples.

6.1 Breast Invasive Carcinoma

This chapter considers a RNA-seq data set[1] of breast invasive carcinoma (BRCA), consisting of $d = 20501$ gene expression measures of $n = 822$ patients. Samples belong to eight different classes, based on their clinical information. Patients are identified based on estrogen receptor positive (ER+) or negative (ER-). Each group is further divided into four classes, based on the lymph node involvement of their disease (N0–N3).

Information on 1859 (human) signalling pathways is extracted from Reactome (Fabregat et al., 2016), a curated and peer-reviewed pathway database.

6.2 Knowledge Driven Network Inference

This learning pipeline consists in two steps:

(a) a pathway selection step, and

(b) a network inference based on the most informative pathways.

Pathways were selected based on their discriminative power for the different classes. The multiclass problem was managed using a one-vs-rest scheme, that is, a binary classification problem for each class against the others. For each pathway, a learning machine consisting of a regularised logistic regression model with ℓ_1 or ℓ_2 penalty (based on the dimensionality of the pathway) was iteratively fit, validated and tested via a model assessment framework based on MCCV (Barbieri et al., 2016). This procedure estimated repeated learning and test scores. The robustness of the system is tested against chance, by means of a random label permutation and re-evaluation of the procedure. The procedure compares the resampled distribution to the random distribution by means of a nonparametric Wilcoxon signed-rank test (Everitt, 1995). Figure 6.1 shows an overview of the pipeline.

The choice of a sparsity-enforcing classifier (via the ℓ_1 penalty) allows to have a list of most selected variables which are important to discriminate output classes. In particular, sparse models assumes that the quantity of interest

[1] Part of TCGA Pancan project, available at https://www.synapse.org/#!Synapse:syn1461151 (last visited: Nov. 21th, 2018).

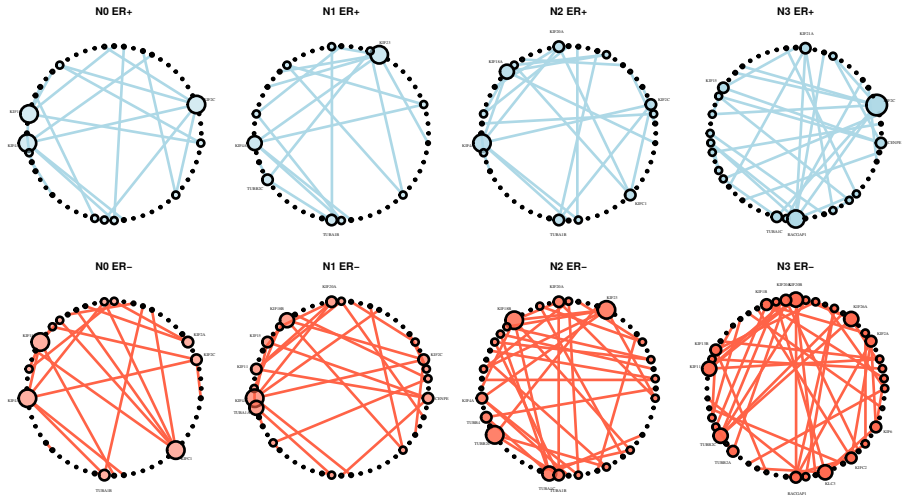


FIGURE 6.2. Network inference on 56 selected variables of Kinesins pathway (R-HSA-983189) for BRCA-affected patients based on the estrogen receptor and lymph node involvement.

depends only on a few relevant variables, which is often the case for biological data. This assumption is at the basis of the construction of interpretable models, since the relevant dimensions allow for a compact, hence interpretable, representation. For pathways which contained less than 100 variables, no feature selection step was employed, and the ℓ_2 penalty was used. The final outcome was predicted as the consensus among all different estimators.

Then, a network for each relevant pathway as selected by this procedure were inferred for each patient, using a pairwise mutual information score. In particular, the network was generated with ARACNE (Margolin et al., 2006) on the expression of genes selected by the estimators. Two nodes are said to interact based on their pairwise mutual information. To avoid an over-representation of the network, edges were limited based on their weights, setting as a threshold the average weight across all edges.

The difference between the networks built on different classes of patients was quantitatively assessed using the normalised Hamming network distance (Hamming, 1950).

6.2.1 Results

Among the 1859 pathways I selected those which obtained the best predictive accuracy and the lowest p-value, indicating the significance of the result and the module under analysis.

This section reports results for the Kinesins pathway (R-HSA-983189), as it harbours groups of genes significantly associated to both lymph node involvement and response to estrogens (Huszar et al., 2009).

Figure 6.4 shows an increasing number of interactions between kinesins from No to N3 lymph node involvement stage, both for ER+ (top row) and ER-

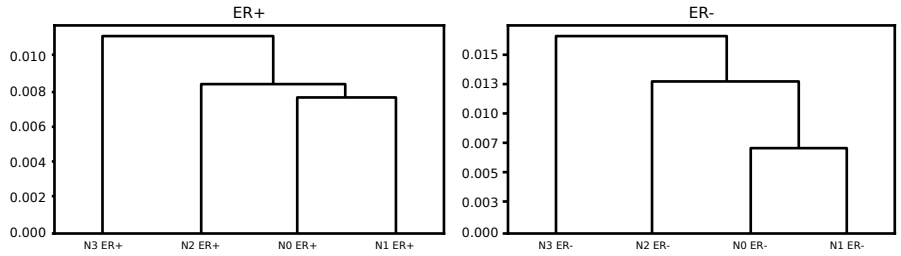


FIGURE 6.3. Network distances based on estrogen receptor and lymph node involvement. Both dendrograms show that the aggregation of consecutive stages occurs sequentially.

(bottom row). On average, graphs associated to ER- have a higher number of interactions with respect to graphs associated to ER+. The distance between networks under different biological conditions were assessed by using the Hamming network distance. In Figure 6.3, a hierarchical clustering algorithm built on such precomputed distances highlights how the network differences are related to the increasing of lymph node involvement considering both ER+ and ER- groups.

Kinesin are microtubule-based motor proteins that mediate diverse functions within the cell, including the transport of vesicles, organelles, chromosomes and protein complexes, as well as the movement of microtubules. These proteins are considered to play a central role in the regulation of mitotic events and potential targets breast cancer therapy, among others (Kaestner and Bastians, 2010). This pathway contains important genes which have been associated to breast cancer, such as KIF20A, KIF20B, TUBB2A and TUBB2C (TUBB4B). The up-regulation of KIF20A, together with its transcription factor FOXM1, is significantly associated with poor survival and with Paclitaxel action and resistance, a chemotherapy medication used to treat breast cancer and other tumour types (Khongkow et al., 2016). Although not as largely characterised, KIF20B share with KIF20A the interactor FOXM1, as experimentally tested. Results on the Kinesin pathway suggests a possible involvement of KIF20B in the malignant progression of breast cancer. The co-occurrence of KIF20A and KIF20B in the N3 ER- graph support this hypothesis.

TUBB2A and TUBB2C genes share an annotation status similar to KIF20A and KIF20B. This means that TUBB2A is more annotated than TUBB2C and the up-regulation of TUBB2A, as well as KIF20A, is known to be significantly associated to Paclitaxel action and resistance (Leandro-Garcia et al., 2012). While poorly annotated, TUBB2C is known to interact with TUBB2A. Results suggests a possible involvement of TUBB2C in the Paclitaxel action and resistance, hypothesis supported by the co-occurrence of TUBB2A and TUBB2C and in the N3 ER- graph.

6.3 *Lasso-based Pathway Selection and Discriminative Analysis*

A limitation of the first pipeline is that there is no straightforward way to assess the pathways as inferred by the resampling strategy. Indeed, the scores are dependent on the number of repetitions of the experiment, and on the metric employed.

This section presents two improvements on the previous pipeline, which aim to reliably select the most informative pathways interpreting the task in a regularised machine learning framework, with the use of group lasso with overlap method. Also, since samples are considered to belong to a common progression trajectory, the network inference should rely on temporal graphical models, such as those described in Chapter 5, with the possibility to quantitatively assess the multiple inferred networks.

6.3.1 Group Lasso with Overlap

Group lasso is a method that is able to select the most relevant variables with respect to a learning goal (Yuan and Lin, 2006). Such norm allows the specification of groups of variables which need to be considered jointly. In fact, the goal is to estimate a sparse set of groups able to discriminate between two or more classes. As introduced in Section 1.2.2, sparsity is a desired property with real data, in particular when the number of samples are low with respect to the variables. Similarly, starting from a potentially high number of groups, the goal of the group lasso penalty is to restrict them based on their discriminative power. When groups are not overlapping, the model assumes the form of (Yuan and Lin, 2006). Instead, in presence of overlapping groups the penalty assumes a slightly different (and more general) form, introduced in (Jacob, Obozinski, and Vert, 2009; Villa et al., 2014).

Consider the general form (1.2). A ℓ_1-norm $\|\boldsymbol{w}\|_1 = \sum_i |w_i|$ leads to sparse models, but does not contain any information on groups of variables which should be selected jointly. Hence, Jacob, Obozinski, and Vert (2009) introduced the group lasso with (possibly) overlapping groups, defined as follows:

$$\Omega_{\mathcal{G}}(\boldsymbol{w}) = \sum_{g \in \mathcal{G}} \left\| \boldsymbol{w}_g \right\|. \tag{6.1}$$

Based on the group lasso with overlapping groups one can use as a prior information a list of pathways, which contains the variables describing BRCA data, as in Section 6.2. Such penalty is based on the ℓ_1-norm, able to enforce sparsity in the solution. As a result, the method extracts a restricted list of pathways, which will be the input for the next step of the second pipeline presented in this chapter.

6.3.2 Discriminant Analysis

Understanding the structure among variables allows to improve the classification task in the case in which the samples belong to different classes. As

such, consider the case where each class of samples have a different covariance matrix, *i.e.*, belongs to a different distribution (see also Section 2.4).

Samples belong to different stages of disease based on their lymph node involvement (Section 6.1). Hence, it is possible to assume underlying similarities between the distributions of the samples. With such ordering, I used the latent variable time-varying graphical lasso method in Chapter 5 to estimate a covariance matrix for each class of samples, where the β parameter is higher than zero, thus exploiting samples of related "neighbouring" classes during the inference of the covariance matrix.

In particular, I used latent variable time-varying graphical lasso twice, for samples having estrogen receptor positive (ER+) and negative (ER-), that is, belonging to separate progression trajectories. The eight covariance matrices are assessed using a single quadratic discriminant analysis, as in Equation (2.13).

Differences between the networks are quantitatively assessed using the weighted spectral distribution (WSD) distance, based on the normalised Laplacian matrix of the networks and the spectral distribution of eigenvalues (Fay et al., 2010).

6.3.3 Results

The classification coefficients, as extracted from group lasso penalty, indicate the best pathways to use for the network inference. Since the penalty is based on the ℓ_1-norm, the number of non-zero elements (that is, the number of groups comprising variables associated to a non-zero weight) is low compared to the number of groups. In particular, given its formulation in Equation (6.1), the group lasso with overlap minimises the number of groups involved in the classification without minimising the non-zero variables *inside* each group (as opposed to sparse group lasso, [Friedman, Hastie, and Tibshirani, 2010]), because of the presence of the ℓ_2-norm.

Table 6.1 contains the list of the pathways as found by group lasso with overlap along with their score, measured as the average weight (in absolute value) assigned to the proteins contained in the pathway for the classification task (discriminating ER+/ER-), and the number of variables included in the pathway.

The pathways selected by the group lasso procedure for the binary classification task (Table 6.1) include R-HSA-983189 (Kinesins), selected as relevant also by the first pipeline. Here it assigned to the last position, with the least average weight as assigned by group lasso. Since this pathway was indeed selected by both procedures, it will be used to compare the following analysis to the first pipeline (Section 6.2). I used the latent variable time-varying graphical lasso on samples belonging to both ER+ and ER- classes, in such a way to maximise the classification score in Equation (2.13).

Figure 6.4 shows the networks for the different classes as estimated by LTGL (6.4a) and ARACNE (6.4b). Then, Equation (2.13) on left out samples (as a test data set) assigned a performance score based on the precision matrices as found by LTGL and ARACNE, respectively. Table 6.4 contains an overview of

TABLE 6.1. Pathway selected by group lasso with overlap (classification task ER+/ER-), with their score and number of proteins associated.

pathway	average weight	cardinality
R-HSA-1251985	9.18e−03	24
R-HSA-383280	7.23e−03	38
R-HSA-1236394	6.74e−03	40
R-HSA-8864260	5.66e−03	35
R-HSA-210500	3.34e−03	23
R-HSA-977068	3.21e−03	21
R-HSA-212676	2.90e−03	22
R-HSA-112310	2.76e−03	50
R-HSA-913709	2.33e−03	60
R-HSA-3906995	2.19e−03	54
R-HSA-5083635	1.72e−03	36
R-HSA-977444	1.54e−03	39
R-HSA-991365	1.54e−03	39
R-HSA-977443	1.46e−03	55
R-HSA-1296059	1.40e−03	25
R-HSA-997272	1.40e−03	25
R-HSA-1296041	1.40e−03	25
R-HSA-419037	6.44e−04	42
R-HSA-983189	2.80e−04	56

the pathways analysed with their multi-class performance metrics (accuracy and F_1-score). The F_1-score is calculated for each label and averaged weighting by support (the number of true instances for each label).

The Kinesins pathway (R-HSA-983189) achieves one of the best classification scores when considering LTGL, as opposed to ARACNE. While both pipelines highlighted such pathway as interesting, Table 6.4 shows the clear advantage in using a temporal graphical model (and also latent factors) during the inference of the networks at each stage of the disease.

Table 6.1 highlights another pathway with the highest weight score, meaning that all of the variables have, on average, the highest weights as assigned by group lasso, namely R-HSA-1251985 (Nuclear signaling by ERBB4). Such pathway was not in precedence selected by the selection procedure described in Section 6.2. A lot of recent studies highlight the relevance of ERBB4 in breast cancer (Chuu et al., 2008; Hollmén et al., 2012; Kim et al., 2016; Sahu et al., 2017; Sundvall et al., 2008), indicating an enhanced ERBB4 processing in breast cancer tissue. Also, empirical observations demonstrate a relation (and coregulation) between estrogen receptor and ERBB4 in breast cancer (Hollmén et al., 2012; Zhu et al., 2006). These observations are in line with the group lasso results, where the task was to discriminate between ER+ and ER- patients. Notably, down-regulation of ERBB4 in ER+ breast cancer cell lines inhibits colony formation, while no effects were observed in ER- cell lines (Tang et al.,

TABLE 6.2. Top 10 pathway selected by group lasso with overlap (classification task of ER+ between N1, N2, N3 and N4), with their score and number of proteins associated.

pathway	average weight	cardinality
R-HSA-977068	1.81e−03	21
R-HSA-3906995	1.45e−03	54
R-HSA-5083635	1.17e−03	36
R-HSA-212676	1.07e−03	22
R-HSA-913709	9.69e−04	60
R-HSA-6794361	8.54e−04	57
R-HSA-442742	7.44e−04	26
R-HSA-112310	7.43e−04	50
R-HSA-442755	7.18e−04	38
R-HSA-438064	6.79e−04	34

TABLE 6.3. Top 10 pathways selected by group lasso with overlap (classification task of ER- between N1, N2, N3 and N4), with their score and number of proteins associated.

pathway	average weight	cardinality
R-HSA-2022090	5.38e−03	59
R-HSA-3906995	4.55e−03	54
R-HSA-8948216	4.41e−03	44
R-HSA-5083635	4.35e−03	36
R-HSA-977068	3.87e−03	21
R-HSA-1296072	3.05e−03	43
R-HSA-2142753	2.89e−03	58
R-HSA-390522	2.67e−03	36
R-HSA-2142691	2.53e−03	21
R-HSA-913709	1.92e−03	60

1999).

Based on R-HSA-1251985 pathway I analysed data with the latent variable time-varying graphical lasso method. Network between proteins in the pathway were validated using the classification score in Equation (2.13). Prediction scores on test data are contained in Table 6.4.

Figure 6.5 shows the networks for the different classes as estimated by LTGL (6.5a) and ARACNE (6.5b). Consistently with the group lasso results, estimated networks by LTGL show visible differences between ER+ with ER- (average WSD network distance 0.079), while differences among the networks belonging to the same ER class are less evident (average WSD network distance 0.074). On the contrary, differences as estimated by ARACNE are negligible in both cases (average distance between ER+ and ER-: 0.007, average distance among ER+ and among ER-: 0.008).

TABLE 6.4. Performance score associated to the precision matrices estimated by LTGL and ARACNE, based on accuracy and F_1-score. The F_1-score is calculated for each label and averaged weighting by support (the number of true instances for each label). The results are computed for one split of the data set. For such particular split, a dummy classifier has F_1-score = 1.8e−01 and accuracy = 1.7e−01 (averaged on 50 repetitions). Hence, ARACNE classification results are below the dummy classifier. Instead, networks as inferred by LTGL can be used to effectively classify test samples.

pathway	ARACNE		LTGL	
	accuracy	F_1-score	accuracy	F_1-score
R-HSA-983189	1.30e−02	3.71e−04	**4.29e−01**	**3.69e−01**
R-HSA-1251985	5.19e−02	3.15e−02	**4.42e−01**	**3.07e−01**
R-HSA-913709	5.19e−02	5.96e−03	**2.73e−01**	**2.35e−01**
R-HSA-445717	1.30e−02	4.40e−04	**4.16e−01**	**3.62e−01**

Table 6.2 highlights another selected pathway that has the highest number of variables and an high weight score, namely R-HSA-913709 (O-linked glycosylation of mucins). Interestingly, recent studies highlights the relevance of R-HSA-913709 pathway in breast cancer, since mechanisms involving proteins in the pathway can contribute to tumour growth and progression (Burchell et al., 2018; Dimitroff, 2015; Mukhopadhyay et al., 2011; Vojta et al., 2016). Such pathway emerges as relevant in all considered tasks, and it has high performance score using LTGL with respect to ARACNE (Table 6.4).

Finally, Figure 6.7 shows the networks for the different classes as estimated by LTGL (6.7a) and ARACNE (6.7b) based on the R-HSA-445717 (Aquaporin-mediated transport). Such pathway did not emerge as significant by the selection procedure described in Section 6.2. Interestingly, recent studies highlights the relevance of R-HSA-445717 pathway in breast cancer since Aquaporins (1–9) have been linked to cancer invasion and metastasis, even though their mechanisms remain unclear (De Ieso and Yool, 2018; Marlar et al., 2017; Satooka and Hara-Chikuma, 2016; Zhu et al., 2018). Recent clinical studies suggested, in particular, the relevance of Aquaporin-3 in tumour progression and prognosis of other malignant cancers, such as colorectal (Li et al., 2013) and hepatocellular carcinoma (Guo et al., 2013). However, the study in this chapter does not include positive and negative samples. Hence, such pathway does not emerge as relevant possibly because diverse genes included in R-HSA-445717 pathway (such as Aquaporins 3 and 5) exhibit expression changes only between carcinoma tissue compared with normal tissue. Nonetheless, using restricting variables to Aquaporins pathway allow to discriminate the different classes of samples available maximising the classification score in Equation (2.13), as shown in Table 6.4. Indeed, the R-HSA-445717 pathway includes lots of genes that are linked to malignant tumour stages. Such genes are seen as markers that is, the stronger is the lymph node involvement, the higher is their inter-dependence, as seen in Figure 6.7. While No and N1 include a small

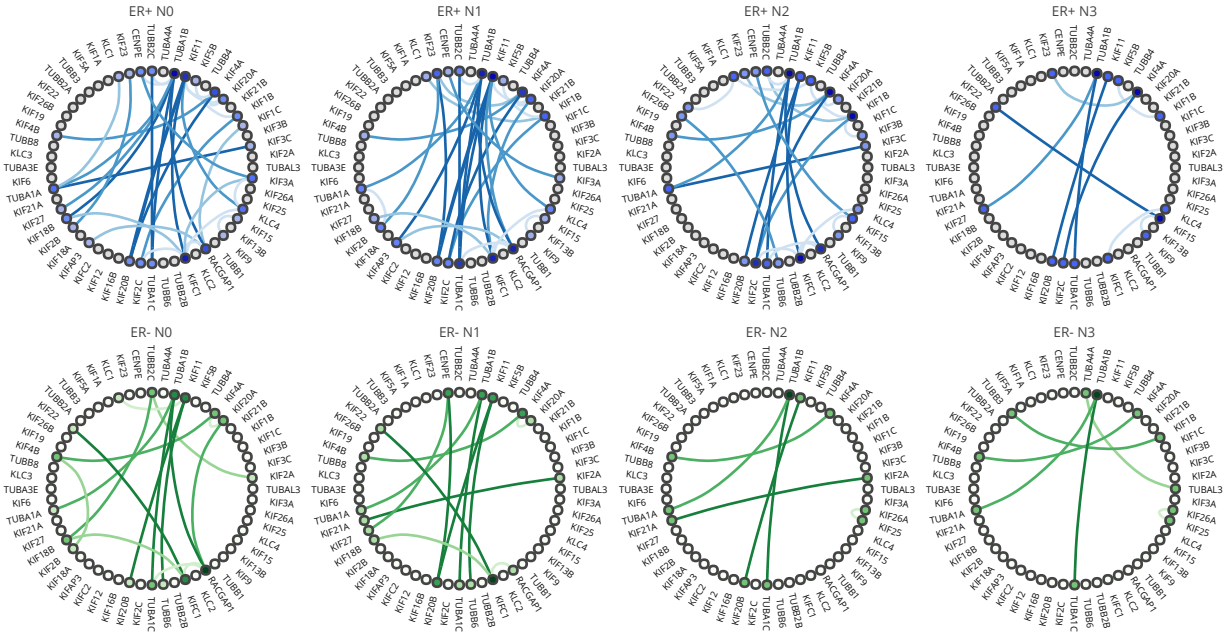
number of links between such genes, the network becomes more connected in N2 and N3.

6.4 Discussion

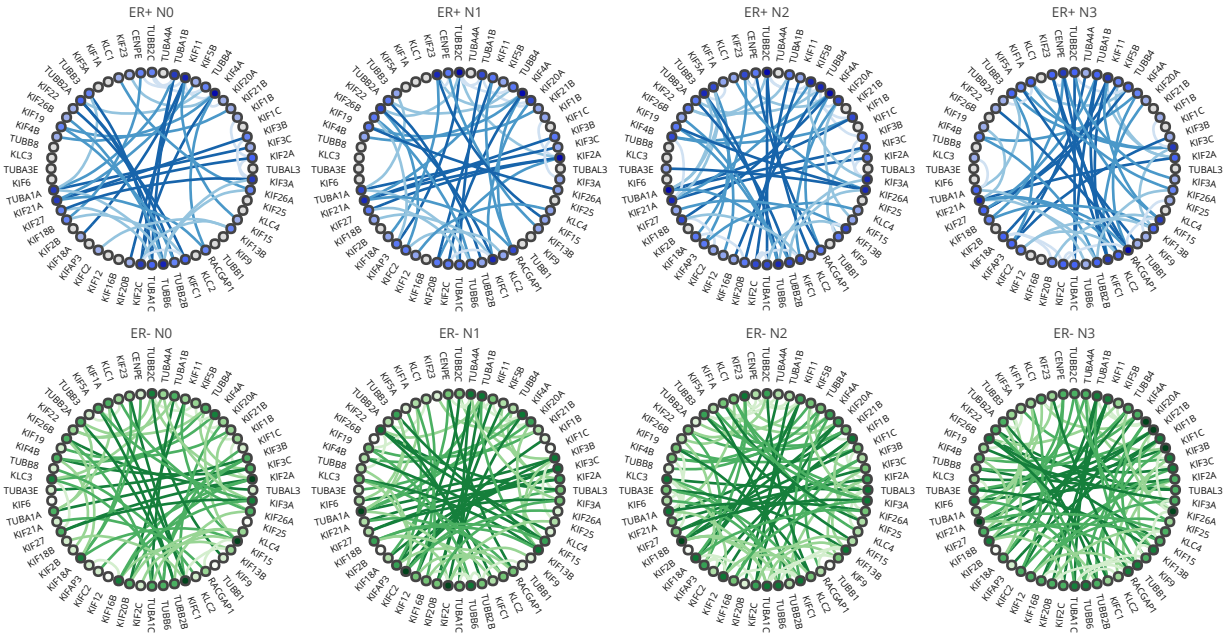
In contexts where the number of samples is low in contrast to the number of variables (a common situation when working with biological data), the use of prior information on the data set at hand *a priori* to direct the analysis represents an effective strategy, instead of relying on it for validation of the results.

The information on the groups of variables (for example, corresponding to nodes of biological pathways) allows to uncover structures under development or evolution. Such differences between stages of a disease may also be quantitatively addressed by the use of network distances or classification scores, based on a discriminant analysis.

Both pipelines described in this chapter allow to reconstruct networks of gene interactions based on different biological conditions. Selecting the relevant modules only partially sheds light on the phenomenon under study. In particular, the interplay among variables within the same module may change when considering different biological conditions. In this context, network inference methods are powerful tools that allow to depict the evolution of the module. The inference of a different network at each stage of the disease, and moreover considering an evolutionary pattern of the variables, paves the way to a more in-depth understanding of how the evolution of the disease affects the interplay among variables involved in specific pathways, which can be effectively interpreted and exploited in clinical contexts.

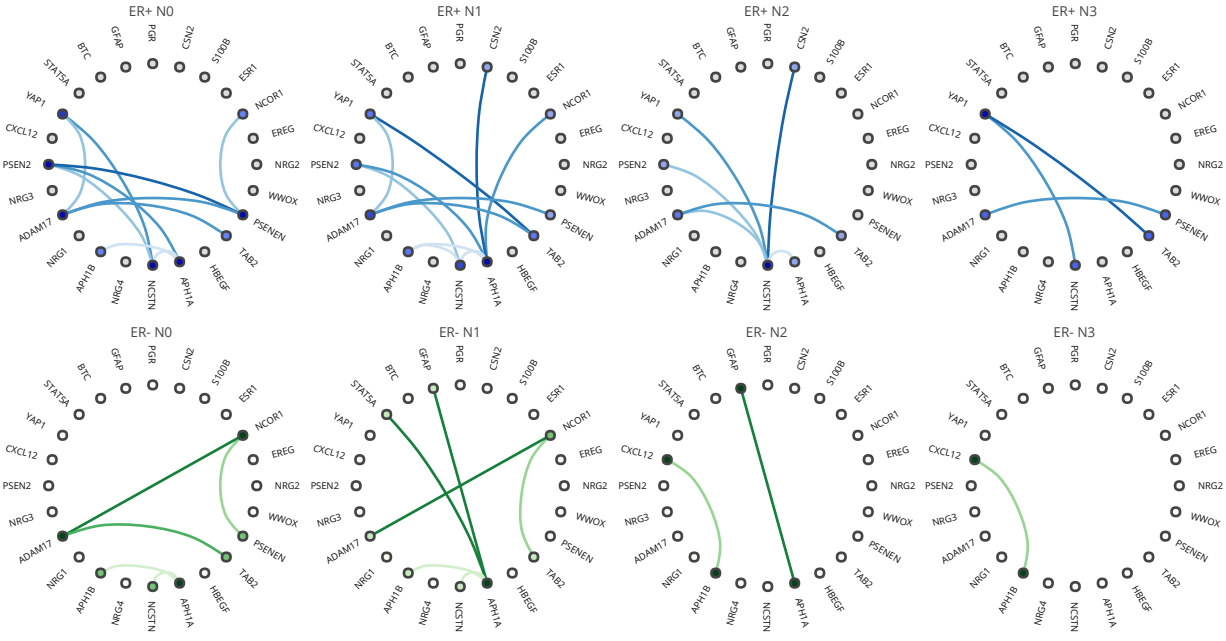


(A) Graphical models for BRCA as estimated by latent variable time-varying graphical lasso.

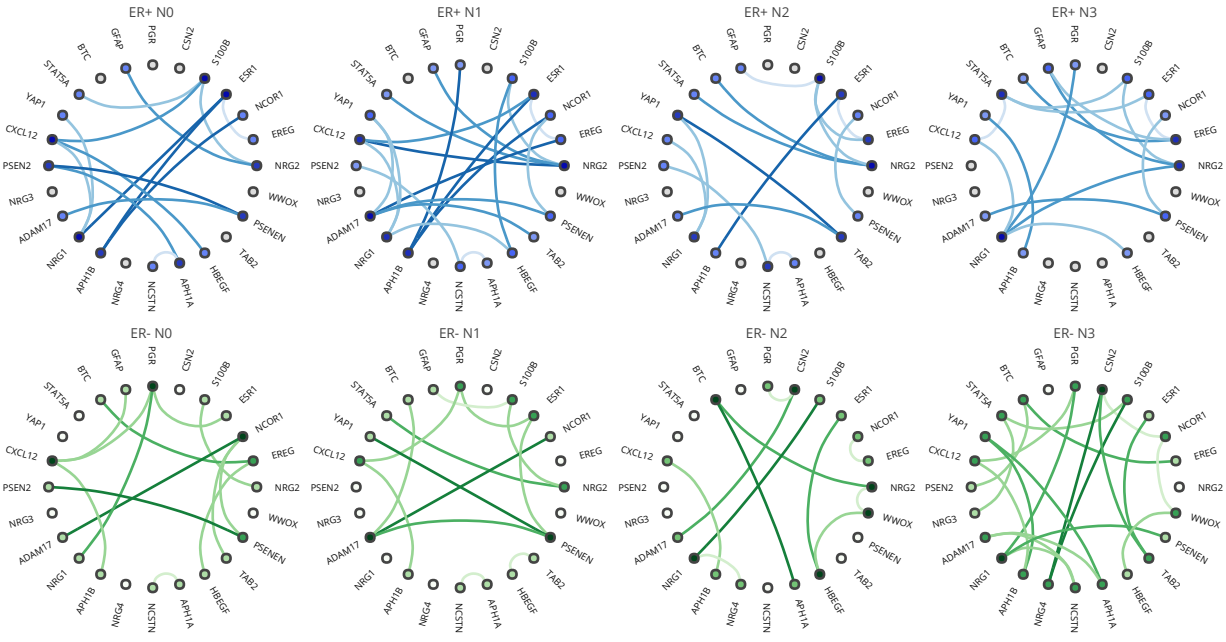


(B) Networks for BRCA as estimated by ARACNE.

FIGURE 6.4. Results for the network inference estimated by latent variable time-varying graphical lasso (6.4a) and ARACNE (6.4b) for the R-HSA-983189 (Kinesins) pathway.

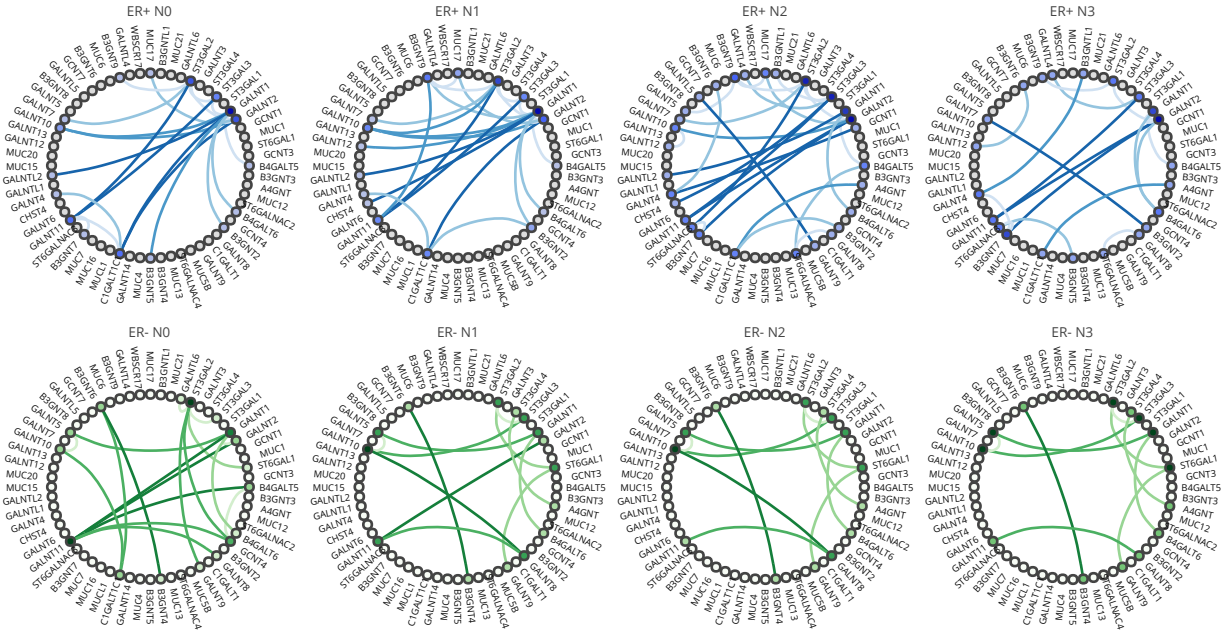


(A) Graphical models for BRCA as estimated by latent variable time-varying graphical lasso.

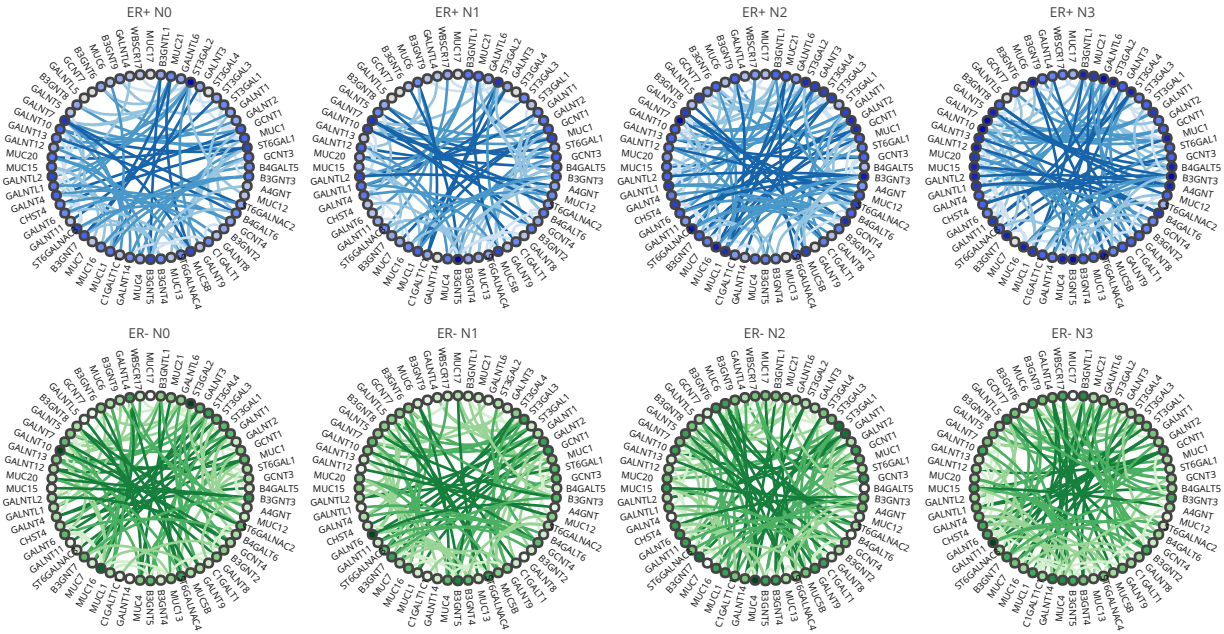


(B) Networks for BRCA as estimated by ARACNE.

FIGURE 6.5. Results for the network inference estimated by latent variable time-varying graphical lasso (6.5a) and ARACNE (6.5b) for the R-HSA-1251985 (Nuclear signaling by ERBB4) pathway.

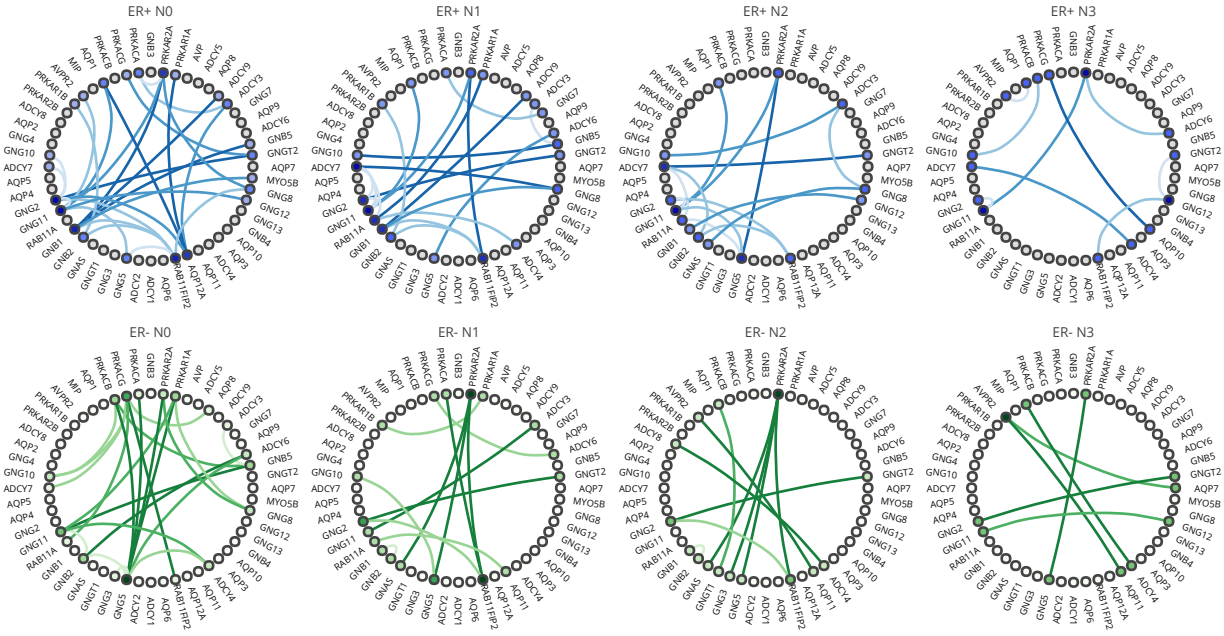


(A) Graphical models for BRCA as estimated by latent variable time-varying graphical lasso.

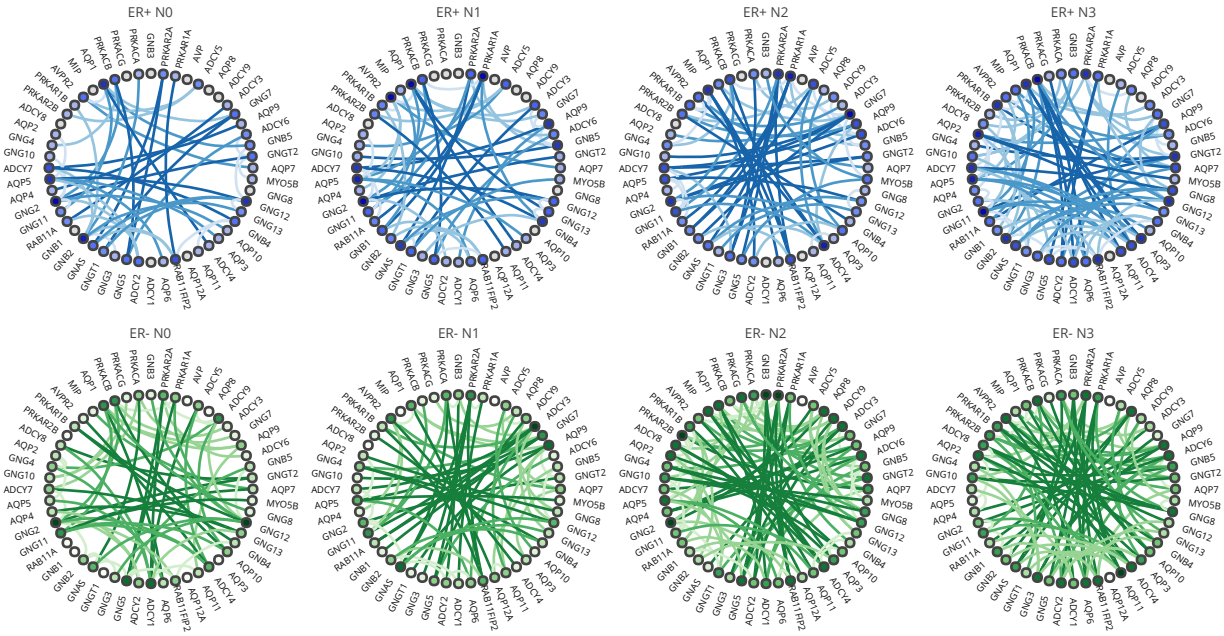


(B) Networks for BRCA as estimated by ARACNE.

FIGURE 6.6. Results for the network inference estimated by latent variable time-varying graphical lasso (6.6a) and ARACNE (6.6b) for the R-HSA-913709 (O-linked glycosylation of mucins) pathway.



(A) Graphical models for BRCA as estimated by latent variable time-varying graphical lasso.



(B) Networks for BRCA as estimated by ARACNE.

FIGURE 6.7. Results for the network inference estimated by latent variable time-varying graphical lasso (6.7a) and ARACNE (6.7b) for the R-HSA-445717 (Aquaporin-mediated transport) pathway.

7 Temporal Models for Single-cell Data

Statistical inference of structure and function of transcriptional regulatory networks is a fundamental and open research question in computational biology. Single-cell sequencing technologies provide a unique opportunity towards such end, given the richness of the data produced. Unlike earlier assays of gene expression measurement, these technologies can expose the heterogeneity within seemingly homogeneous groups of cells helping to infer the regulatory mechanisms of transcription in multicellular organisms, a crucial task to understand and model cellular processes. This chapter presents a possible application of latent variable graphical models for transcription regulation using scRNA-seq data, including temporal information regarding cell evolution through the use of appropriate priors. Single-cell data present challenges due to their nature. In fact, such data are highly-dimensional, intrinsically sparse and subject to high levels of noise. Hence, a graphical modelling of single-cell data without sparsity-enforcing priors is not feasible. The application of time-varying graphical lasso and latent variable time-varying graphical lasso (Chapters 4 and 5) inherently include both a sparsity prior on single networks (at each time point) and a temporal component, to constrain and overcome the disadvantages of working with such sparse and noisy data, providing a useful framework for graphical modelling of single-cell data.

Outline

The rest of the chapter is organised as follows. Section 7.1 introduces single-cell sequencing, a recent and powerful technology for sequencing biological data. Section 7.2 describes the data used in the following analysis, *i.e.*, haematopoietic stem cells. Section 7.3 includes the analysis involving latent variable time-varying graphical lasso based on a pseudotemporal ordering of the cells. Section 7.4 concludes with an overview of the work and further research directions.

7.1 Single-cell Data

Single-cell sequencing is a recent approach to characterise gene expression at the single cell level (Tang et al., 2009) and it has opened the possibility to investigate cellular heterogeneity in terms of RNA expression, protein abundance and metabolites (Blainey and Quake, 2014; Sandberg, 2014; Spitzer and Nolan, 2016; Zenobi, 2013). In particular, the single-cell sequencing technology allows to separately sequence each cell, thus having the expression information at the particular state of the cell during its evolution.

Standard expression experiments (bulk RNA-seq) are limited in the sense that they provide measurements that result from the averaging of heterogeneous populations of cells, thus masking or smoothing the signal of interest. Single-cell data, instead, provide targeted measurements that characterise each cell differently. In this way heterogeneous measurements can be uncovered, with increasing variability and expression distributions. Particularly, molecular information at the resolution of single cells allows to investigate cellular diversification, a key factor to understand the underlying complexity of different cell states (Yuan et al., 2017).

Common goals with such data include the identifications of subpopulations of cells within a particular biological context, the characterisation of differentially distributed genes across cells and conditions, and pseudotime reconstruction (Bacher and Kendziorski, 2016). The latter is crucial to interpret the exact state of a cell at a fixed time. The pseudotime reconstruction is needed because single-cell profiling happens after the cell has been isolated from its local environment and destroyed. While informative for the expression levels of genes and proteins of the cell, the spatial-temporal context of the cell itself at the moment of the sequencing is lost. Hence, lots of methods aims to infer both the spatial environment of a cell and its state in a trajectory of dynamic behaviour using the measured gene expression (Skylaki, Hilsenbeck, and Schroeder, 2016). Single-cell data have the great advantage not to obscure or misrepresent the signal of interest (Trapnell, 2015). Such data, while offering opportunities to discover molecular patterns that were hidden in traditional expression experiments, pose in contrast challenges for standard statistical and computational methods.

This chapter focuses on single cell RNA sequencing (scRNA-seq), one of the most important tool for single-cell analysis (Skylaki, Hilsenbeck, and Schroeder, 2016). While having the mentioned advantages over earlier expression measurement techniques, such as RNA-seq (bulk), scRNA-seq data analysis is challenging as the data produced is intrinsically noisier, due to the low amount of starting material coupled with sparse sampling (Yuan et al., 2017), thus not allowing the intrinsic noise attenuation as in bulk RNA-seq. However, such data may allow the regulatory network inference using the variation across cells, taking into account the variation caused by the longitudinal information of the cell itself.

7.2 Haematopoietic Stem Cell Development

The entire blood system can be restored from a single haematopoietic stem cell (HSC), which makes the HSC transplantation a therapeutic option. Manufacturing HSCs in the laboratory, *i.e.*, a *de novo* generation of haematopoietic stem and progenitor cells would constitute a powerful treatment of blood disorders. However, the derivation of HSCs from pluripotent stem cells has not yet been fully understood (Wahlster and Daley, 2016). In particular, attempts were made to reprogram non-haematopoietic cell types into HSCs, but these efforts have not been successful (Doulatov et al., 2013; Lis et al., 2017). A possibility to

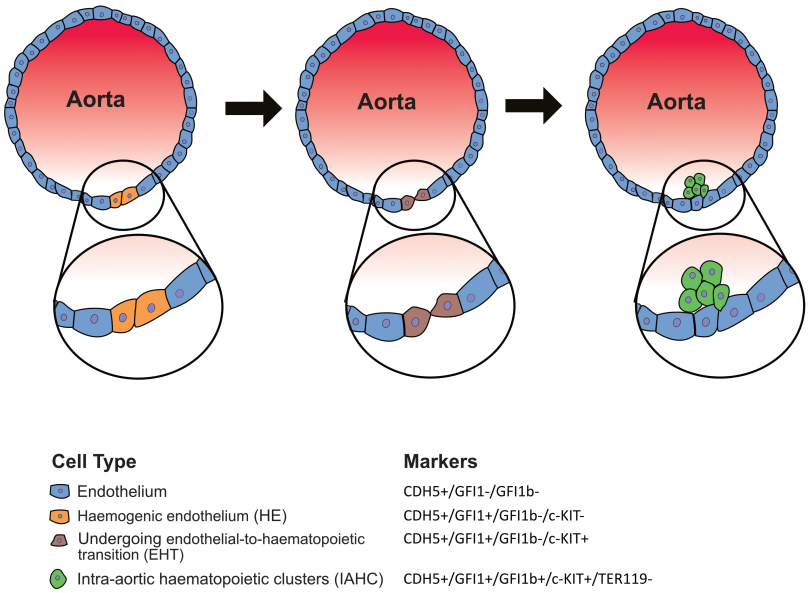


FIGURE 7.1. Generation of haematopoietic stem cell.

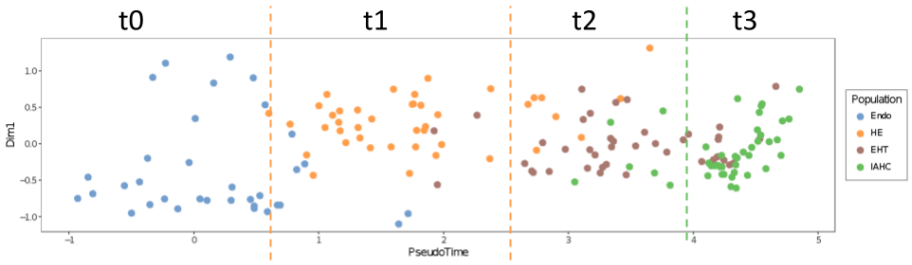


FIGURE 7.2. Pseudotime ordering.

convert human adult endothelial cells into transplantable multipotent haematopoietic progenitors have been possible with the involvement of transcription factors FOSB, GFI1, RUNX1, and SPI1 (Sandler et al., 2014), under important limitations (Lis et al., 2017).

The following analysis involved single-cell data composed by 152 haematopoietic (Mus Musculus) stem cells, divided into four pseudotime points by GrandPrix, a Bayesian Gaussian process latent variable model (GPLVM) able to reduce single-cell gene expression profiles into a low-dimensional space (Ahmed, Rattray, and Boukouvalas, 2017).

The pseudotemporal ordering assumes cells to represent a time-series, where each cell belongs to a particular time point in a pseudotime trajectory. Such trajectory corresponds to a process of interest where, in the case of haematopoietic stem cell, this is represented by the development and specialisation of the cell. Figure 7.2 shows the low-dimensional representation of the cells. Consistently to their trajectory, the pseudotime ordering shows the progression from endothelium to intra-aortic haematopoietic cell type, corresponding

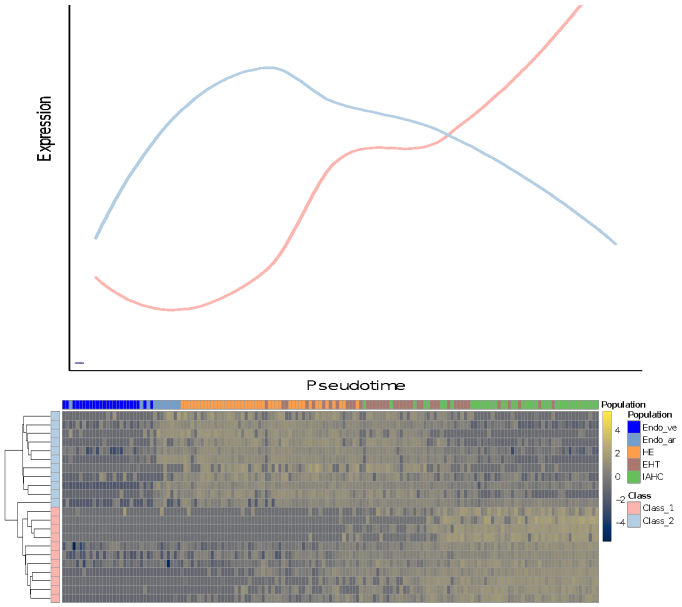


FIGURE 7.3. Evolution of the expression of two classes of genes. The gene behaviour across cells reflects the division into four pseudotemporal steps.

to the final step in their trajectory.

Figure 7.3 shows how the cells, divided into the four pseudotime steps, exhibit consistent developmental progression across the samples.

7.3 *Network Inference*

Based on the pseudotime ordering, cells are grouped into developmental states. Next, I used the latent variable time-varying graphical lasso to infer the dynamical network across the four different pseudotime points. For computational advantage, 2444 gene expression levels were considered.

I devoted particular attention on the β penalty across the difference between time points (Section 5.1). Based on the pseudotemporal ordering, the temporal penalty of precision matrices $\boldsymbol{\beta} = (\beta_{0-1}, \beta_{1-2}, \beta_{2-3})$ is considered as a vector to enforce different temporal similarities.

In particular, β_{0-1} was set to 0, since the different cells at $t = 0$ may follow different evolutionary trajectories. Instead, β_{1-2} and β_{2-3} have strictly positive values, since cells at time $t = 1$ are bound to develop into endothelial-to-haematopoietic transition (EHT) and intra-aortic haematopoietic clusters (IAHC) stages. Table 7.1 shows the top five interactions across the time points. Such interactions involve EIF2S3Y, DDX3Y and XIST, shown to be sex-chromosome linked and expressed in the neonatal mouse heart (Ehmann et al., 2008; Isensee et al., 2008). Figure 7.4 shows a gene network based on the Runt Related Transcription Factor 1 (RUNX1) gene, a transcription factor fundamental in haematopoietic stem cell development (Sugimura et al., 2017). Table 7.2 shows the top five pathways after the enrichment based on the biological progresss (GO) and KEGG pathways. The haematopoietic pathways

gene pair	t_{0-1}	t_{1-2}	t_{2-3}
HSPA1A–HSPA1B	−6.28	−6.25	−6.22
XIST–EIF2S3Y	5.38	5.36	5.33
XIST–DDX3Y	4.54	4.53	4.49
CCNA2–UB92C	−4.01	−3.98	−3.93
RPS2-PS10–XIST	−3.93	−3.99	−3.99

TABLE 7.1. Relevant interactions across time.

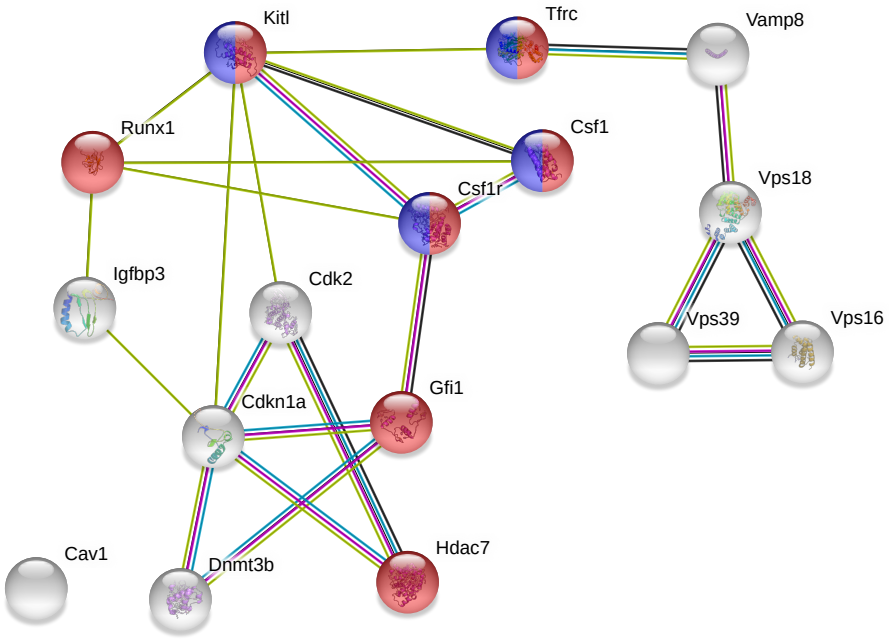


FIGURE 7.4. Gene network based on Runt Related Transcription Factor 1 (RUNX1) gene. Colors of nodes refer to pathways in Table 7.2, while colors of edges refer to the specific types of interactions between genes. Image generated with https://string-db.org.

are the most relevant among both GO and KEGG pathways, with a low false discovery rate (order of magnitude of 10^{-5} and 10^{-4}).

Figure 7.5 shows the inferred networks for each time step. While most of the nodes are not connected, nodes in the centre of the layout became progressively more connected at $t = 2$ and $t = 3$.

7.4 Discussion

The advent of scRNA-seq technology brought new challenges along with an incredible source of potential information. Currently, due to the novelty of such a technology, methods to efficiently deal with scRNA-seq data are scarcely available. Inference of regulatory networks is an open question in biomedical data science. With the addition of longitudinal information of the network,

	pathway ID	pathway description	gene count	false discovery rate
Biological Process (GO)	0046718	viral entry into host cell	4	5.59e−06
	0030097	hemopoiesis	7	5.22e−05
	0002684	positive regulation of immune system process	7	6.95e−05
	0002763	positive regulation of myeloid leukocyte diffe…	4	6.95e−05
	0002520	immune system development	7	7.45e−05
KEGG Pathways	4640	Hematopoietic cell lineage	4	7.95e−05
	4151	PI3K-Akt signaling pathway	5	2.74e−04
	5200	Pathways in cancer	5	2.74e−04
	5202	Transcriptional misregulation in cancer	4	2.75e−04
	4115	p53 signaling pathway	3	7.30e−04

TABLE 7.2. Relevant pathways after enrichment process on a subset of the inferred network.

the task becomes quickly intractable for state-of-the-art methods for graphical modelling.

Novel statistical methodologies are required to tackle the problem of transcriptional regulatory network inference. In this context, the latent variable graphical modelling using scRNA-seq data after a pseudotime ordering offers a great opportunity to discover novel as well as established interactions between genes. In particular, the method is designed to model scRNA-seq data at each time-step as a network with latent variables and simultaneously relate networks close in time. The inferred networks are validated by showing the consistency with the underlying biology. Further validation may offer novel insights into the regulatory network structure of multicellular organisms.

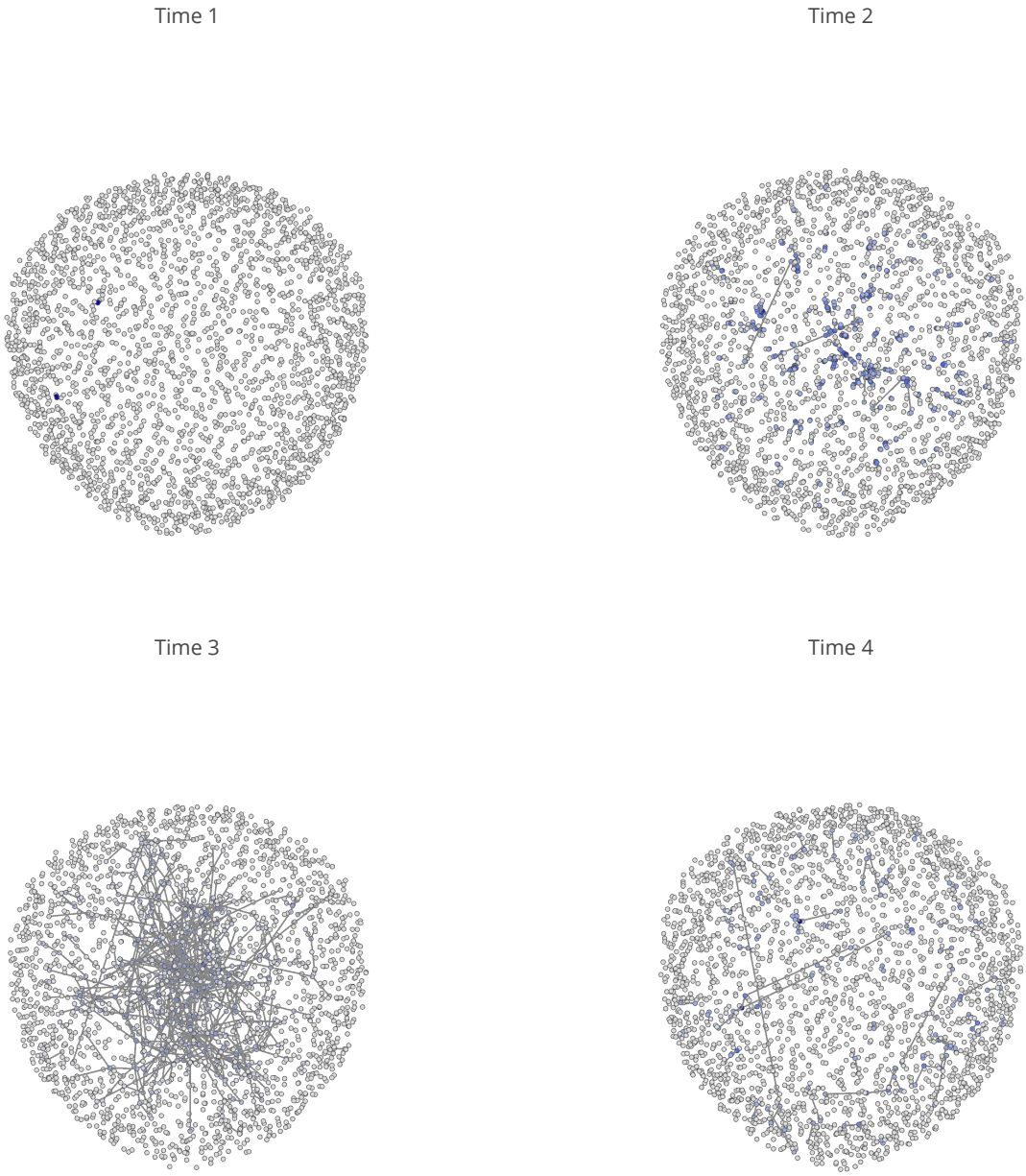


FIGURE 7.5. Visual representation on the global inferred network after LTGL, for $t = 1, 2, 3, 4$.

8 Wishart Process for Epilepsy

Estimating a series of covariance matrices indexed by time is daunting task. Chapter 4 introduces the task of graphical modelling of time-series under a sparsity constraint which allows the inference of a wide set of variables based on a small number of samples, which would normally not be possible under standard statistical guarantees. The time-varying graphical lasso is one of the state-of-the-art methods to both incorporating temporal dynamics across covariance matrices over time and enforcing sparsity in each single precision matrix. However, TGL and LTGL introduce an approximation following the discretisation of the time-series.

A similar approach interprets the sequence of covariance matrices over time as a single draw from a generalised Wishart process (GWP), *i.e.*, a process with Wishart marginals (Cardona, Álvarez, and Orozco, 2015; Fox and West, 2011; Wilson and Ghahramani, 2011). This chapter introduces an application for Wishart process for epilepsy data, following the analysis in (D'Amario et al., 2018). Such method allows to natively consider the temporal evolution of the epileptic channels, to understand how correlation between channels evolves over time.

Motivation

Epilepsy is a neurological disorder affecting more than 50 million people worldwide. This disease is characterised by abrupt loss of consciousness and convulsions, causing severe impairments in daily life. The occurrence of epileptic symptoms can be local (focal seizure) or general (general seizure). In the first case, the seizure onset zone is restricted to a portion of the brain which produces both hyper synchronisation and hyper activity typical of the pathology (Jiruska et al., 2013). The onset zone can be further sub-categorised into *(i)* epileptogenic areas, which generate the epileptic activity and *(ii)* irritative areas that actively contribute to the propagation. In the rest of the chapter such areas will be generally referred to as *critical* (or, equivalently, pathological).

About 30% of focal epileptic patients do not respond to pharmacological treatments, needing surgical ablation of the pathological area. In such cases, the identification of the minimal amount of neural cortex to ablate for seizure-free outcomes, namely the epileptic zone (EZ), is an extremely delicate and precise task. To this aim, clinicians use non-invasive methodologies such as magnetic resonance imaging, computed tomography and scalp electroencephalography as first clinical tests, seeking for clear evidence of tumours or dysplasia which may cause the seizures. Nonetheless, EZ borders may be difficult to localise. Medical experts often resort to the use of invasive investigation techniques such as stereo-electroencephalography (SEEG) to assess critical areas.

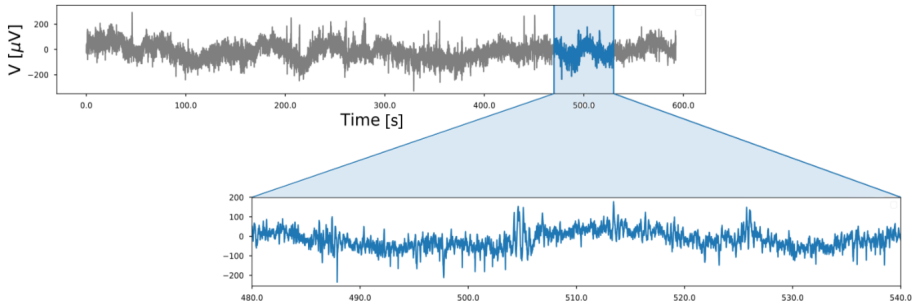


FIGURE 8.1. Example of epileptic signal corresponding to 10 minutes of acquisition. SEEG recordings are characterised by high sampling frequencies (1 kHz). These signals are usually analysed by clinical experts that look for biomarkers, a subjective and error-prone process.

SEEG measures the electrical activity from intracranial areas through filiform electrodes implantation, where each sensor is endowed with dozen of acquisition channels. These recordings have high spatial and temporal correlation (Figure 8.1), because of the complexity of the brain structure and the long period acquisitions at high sampling frequencies ($\geq$ 1 kHz), respectively.

Related Work

The characterisation of SEEG signals is a challenging and time-consuming task, usually based on visual inspection or signal processing tools, and it is intrinsically subjective, possibly leading to misclassification (Soriano et al., 2017; Staba, Stead, and Worrell, 2014; Yardi et al., 2016). Therefore, automatic classification of neural recordings is an emerging field. In this context, methods consider both temporal and spectral representation of the signal (Omerhodzic et al., 2013). The signal can in fact be described by its energy at different frequency bands, which shows highest discriminative power in seizure onset zone detection (γ frequency band, 20–70 Hz) (Vila-Vidal et al., 2017). Also, the quantification of energy concentration at different bands can be a measured with wavelet entropy (Mooij et al., 2016; Rosso et al., 2001).

Another typical approach for the classification of epileptogenic channels is through the definition of the most informative biomarkers. Interictal spikes and spike-and-wave complexes are considered a well established evidence of the pathological condition (Avoli, Biagini, and De Curtis, 2006; Curtis and Avanzini, 2001).

High-frequency oscillations (HFOs) are short events (2–5 ms) at frequencies in the range of 80–500 Hz, sub-categorised in *ripples* (80–200 Hz) and *fast ripples* (200–500 Hz). HFOs are considered as a good predictors for EZ localisation (Fedele et al., 2017). The role of HFOs in seizure generation has been object of investigation, which attest reliable co-occurrence of HFOs in critical areas (Jacobs et al., 2012). Indeed, several works reveal the primary role played by HFOs as biomarkers for epilepsy (Höller et al., 2015; Zijlmans et al.,

2012).

Contribution

Subject to high level of intrinsic correlation, understanding the evolving relations between channels is not a trivial task. Indeed, a quantitative measure of correlation between channels is a desirable result to compare different brain areas in focal epilepsy. This would consistute an appropriate application for the latent variable time-varying graphical lasso method, which would aim at reducing spurious signal correlations. However, the application of sparse graphical models introduces a significant coarse approximation in the time-series, in particular when the temporal window is highly restricted.

Also, time-series may exhibit complex relation patterns, which may involve time points that are not necessarily contiguous. In that case, instead of a penalty across subsequent time points, a reasonable approach would be to consider a *temporal kernel* across all time points, allowing for the possibility to flexibly model generic complex interaction patterns that may be expressed via kernels.

This final chapter presents a different approach for inferring developing relations among time-series variables when sparsity is not a necessary property of the estimated graphical model, integrating the work on the graphical models for temporal data developed throughout this thesis.

Outline

The rest of the chapter is organised as follows. Section 8.1 briefly details the data used. Section 8.2 presents the data analysis pipeline developed for the analysis of stereo-electroencephalography (SEEG) data through a multi-task multiple kernel learning approach. Section 8.3 introduces the Wishart process and an algorithm for their inference. Section 8.4 contains the results of the Wishart process application on the SEEG data. Section 8.5 concludes with an overview of the pipeline explained in this chapter.

8.1 Stereo-electroencephalography Time-Series

Data comprise signals recorded from 18 patients, acquired at Ospedale Ca' Granda Niguarda, Milan (Italy)[1] with the acquisition system of Arnulfo et al. (2015).

Each electrode was endowed with a varying number of channels (8–15). For each patient, data included 590 seconds of spontaneous interictal activity with closed eyes at resting state, at 1 kHz sampling frequency, comprising a total number of channels of 2347, 984 of which tagged as critical by medical experts.

[1]Patients provided written consent for further analysis with scientific research purpose.

8.2 Multiple Kernel Learning for Epilepsy

The localisation of epileptic zone in pharmacoresistant focal epileptic patients is a difficult task, typically performed by medical experts through visual inspection over neural recordings. For a finer localisation of the epileptogenic areas and a deeper understanding of the pathology both the identification of pathogenical biomarkers and the automatic characterisation of epileptic signals are desirable. This section presents a data integration learning method based on multi-level representation of stereo-electroencephalography recordings and multiple kernel learning. To the best of the author's knowledge, this represents a first attempt to tackle both aspects simultaneously, as the approach described in what follows is devised to classify critical and non-critical recordings while detecting the most discriminative frequency bands.

Indeed, this section introduces a machine learning method which simultaneously tackles the problem of searching for informative frequency bands and localising critical areas in focal epilepsy, through a multi-scale integration of SEEG recordings. The method leverages on continuous wavelet representation of the signal, exploiting its multi-level nature to obtain a redundant description that is integrated through pairwise similarity measures in the learning pipeline.

Since the number of channels and the type of signal acquired changes across patients, a direct comparison between patients is not feasible. The proposed procedure overcomes this issue by extending a multiple kernel learning (MKL) algorithm in a so-called multi-task multiple kernel learning (MT-MKL) (Borgwardt, 2011; Wang et al., 2010). MT-MKL aims at optimising a multi-task classification problem. It includes additive constraints guaranteeing robustness to noise, providing interpretable and stable results. The outcome of the method incorporates both a personalised description for each patient and the selection of the best descriptors of the pathology across the population.

8.2.1 Data Representation through Multi-Scale Analysis

A multi-scale representation aims at differentiating a signal in several frequency bands, each corresponding to a different behaviour of the recording. Indeed, the wavelet transform offers a reliable tool that, due to its local nature, is able to detect transients through a time-frequency representation. In particular, the continuous wavelet transform (CWT) with a set of generators (mother wavelet) gives a rich and redundant decomposition of the signal (Mallat, 1999).

The mother wavelet is defined to be the complex Morlet transform:

$$\psi_{\tau,s}(t) = \frac{1}{\sqrt{\pi s}} e^{2i\pi \frac{t-\tau}{s}} e^{-\frac{(t-\tau)^2}{s^2}}, \tag{8.1}$$

where τ, s denote respectively the temporal shift and the scaling parameter. For any one dimensional signal $\boldsymbol{x}(t)$, its representation through wavelet transform ψ at a fixed scale s is given by the coefficients $W_{\tau,s}(\boldsymbol{x})$. Hence, CWT results in a two dimensional representation of the original signal.

8.2.2 Similarity Measures

The decomposition of neural signals through CWT is necessary in the perspective of a multi-level pairwise comparison and a deeper insight in the role played by different frequency rhythms in the discrimination of EZ. For a comprehensive description of the signal, it is necessary to consider measures involving both phase and amplitude correlation, taking into account the possible temporal shift between recordings. Indeed, signals were encoded into kernels via phase locking value (PLV), normalised correlation and spectral measures, computed for each scale s of the wavelet under analysis (D'Amario et al., 2018).

8.2.3 Multiple Kernel Learning

Multiple kernel learning (MKL) integrates data by combining sets of kernel functions (Borgwardt, 2011; Lanckriet et al., 2004). Kernels are positive semi-definite matrices whose entries $K_{ij} = \kappa(x_i, x_j)$ encode pairwise similarities between data points (x_i, x_j). However, the choice of the most suitable kernel function for each problem at hand is tricky, and heavily depends on available data. Therefore, the idea behind MKL is to construct different measures of similarity on the same data set and then integrate them into a single kernel (Gönen and Alpaydın, 2011). For example, a straightforward MKL may be a linear combination of different kernels (Borgwardt, 2011). This is possible given the fact that kernels allow linear operations while preserving their mathematical properties, such as positive semi-definiteness and symmetricity (Friedman, Hastie, and Tibshirani, 2001). Formally, consider k kernels $\{K_1, \ldots, K_k\} \in \mathbb{R}^{k \times n \times n}$ that represent different similarities measures among points of a data set. Kernels can be combined linearly as a weighted sum $\sum_{i=1}^{k} w_i K_i$, where $\boldsymbol{w} = (w_1, \ldots, w_k)^\top \in \mathbb{R}_+^k$ is a list of (non-negative) coefficients, measuring the relevance of each kernel for the particular problem at hand.

8.2.4 Multi-Task Multiple Kernel Learning

Implantation settings strongly depend on the patient clinical condition and on preliminary medical evaluations, based on previous non-invasive clinical tests. Such variability does not allow for a direct comparison of the neural activity across patients. In other words, the different acquisition procedure for each patient denies a single unified regression model for all patients. For this reason, the analysis pipeline extends the MKL to account for different patient conditions, resulting in multi-task multiple kernel learning (MT-MKL). Each kernel represents a particular similarity matrix among all the channels in a single patient at a specific scale. The innovation of such method consists in the capability of jointly analysing the patients by taking into account their diversity.

The goal is three-fold: *(i)* to combine kernels to predict whether a channel is epileptic or not, *(ii)* to identify the most informative kernels for prediction across patient, and *(iii)* to select relevant channels for each patient. After the preprocessing step, data for each patient consists in a matrix $X^{(p)} \in \mathbb{R}^{c_p \times T}$ and a vector of labels $\boldsymbol{y}^{(p)}$ denoting the pathological or physiological nature of each signal in $X^{(p)}$. Note that the number of channels c_p varies across patients, and the proportion of epileptic channels is not uniform across the population. Let $(K_1^{(p)}, \ldots, K_k^{(p)})$, where $K_j^{(p)} \in \mathcal{S}_+^{c_p}$, be the set of k kernels for a patient p. The decision function $f^{(p)}$ for a patient p and a channel x is defined as:

$$f^{(p)}(x) = \alpha_0^{(p)} + \sum_{i=1}^{c_p} \left[\alpha_i^{(p)} \sum_{j=1}^{k} w_j K_j^{(p)}(x_i, x) \right], \tag{8.2}$$

where $\alpha_i^{(p)}$ denotes the i-th component of the regression parameter $\boldsymbol{\alpha}^{(p)}$ specific for each patient p. Having separate parameters ($\boldsymbol{\alpha}^{(p)}$ and $\boldsymbol{w}$) is fundamental for the resolution of the problem. In fact, $\boldsymbol{\alpha}^{(p)}$ allows to better approximate the labels $\boldsymbol{y}^{(p)}$ by capturing the variance of each patient, while $\boldsymbol{w}$ combines the kernels by weighting them and, as it holds across patients, indicates the most discriminative kernels.

The addition of a $\ell_1\ell_2$ penalty on $\boldsymbol{w}$ and $\boldsymbol{\alpha} = (\boldsymbol{\alpha}^{(1)}, \ldots, \boldsymbol{\alpha}^{(n)})$, also, ensures interpretable and stable solutions. By considering all the patients, our goal translates into minimising the following objective function:

$$\begin{aligned} \underset{\boldsymbol{\alpha}^{(1)},\ldots,\boldsymbol{\alpha}^{(n)},\boldsymbol{w}}{\text{minimize}} \Bigg\{ & \sum_{p=1}^{n} \left(\ell_{f^{(p)}} \left(X^{(p)}, \boldsymbol{y}^{(p)} \right) + \lambda (r_\lambda \|\boldsymbol{\alpha}^{(p)}\|_1 + (1 - r_\lambda) \|\boldsymbol{\alpha}^{(p)}\|_2^2) \right) \\ & + n\beta \left(r_\beta \|\boldsymbol{w}\|_1 + (1 - r_\beta) \|\boldsymbol{w}\|_2^2 \right) \Bigg\} \\ & \text{s.t. } w_j \geq 0 \text{ for each } j = 1, \ldots, k \end{aligned} \tag{8.3}$$

where $\ell_{f^{(p)}} \left(X^{(p)}, \boldsymbol{y}^{(p)} \right) = -\sum_{i=1}^{c_p} \log(1 + \exp(-y_i^{(p)} f^{(p)}(x_i)))$ is the negative log-likelihood of the logistic probability function and r_λ and r_β are the elastic-net penalty ratios on $\boldsymbol{\alpha}$ and $\boldsymbol{w}$, respectively. The elastic-net penalty benefits from the stability property of the ℓ_2 regularisation term (Zou and Hastie, 2005).

8.2.4.1 Minimisation Method

The optimisation of Equation (8.3) relies on alternating minimisation (Bolte, Sabach, and Teboulle, 2014). Note that the problem is not jointly convex in both $\boldsymbol{\alpha}$ and $\boldsymbol{w}$. Hence, no theoretical guarantees on convergence to a global minimum exist. Problem (8.3) is bi-convex — *i.e.*, it is convex in each variable keeping the other fixed. Its optimisation is based on an alternating forward-backward splitting procedure given the non-differentiability of parts of the functional (ℓ_1 norm) (Bolte, Sabach, and Teboulle, 2014; Combettes and Vũ, 2014). Algorithm 4 describes the optimisation procedure.

Algorithm 4: Alternating minimisation algorithm for the MT-MKL.

```
Initialise α^(1)(0), ..., α^(n)(0), w(0);
for t < t_max do
    for p = 1, ..., n do
        α^(p)(t) ← minimise Problem (8.3) with w = w(t − 1)
    w(t) ← minimise Problem (8.3) with α = α(t);
    if stop criterion is met then
        return α^(1)(t), ..., α^(n)(t), w(t)
```

8.2.4.2 Minimisation of $\boldsymbol{\alpha}$

Fixing $\boldsymbol{w}$, for each patient p the functional with respect to $\boldsymbol{\alpha}^{(p)}$ takes the form of a standard logistic regression. Its minimisation is performed by computing the derivative on the logistic loss and then applying the soft-thresholding operator (Tibshirani, 1996) on the result of the gradient descent step.

8.2.4.3 Minimisation of $\boldsymbol{w}$

The minimisation of $\boldsymbol{w}$ is non-separable across patients. Its gradient, computed on the differentiable part of the functional, is a sum of gradients computed for each patient p. Then, the soft-thresholding operator is applied to enforce sparsity in the solution. Also, kernel weights are projected into the positive half-space by applying a threshold on zero. This ensures that a kernel is considered only if its weight is positive, otherwise it is discarded.

8.2.5 Pipeline Design

The proposed pipeline consists of three main steps: *(i)* signal preprocessing and multi-scale representation of the signals, *(ii)* computation of similarity measures, and *(iii)* learning the optimal combination of kernels and channels weights for signal classification (Figure 8.2).

In step *(i)*, given an input matrix $X^{(p)}$ we re-refer the potential using the bipolar montage, which consists in the differential measures between two adjacent channels. Local reference is standard for phase measures, as it reduces volume conduction effects caused by white matter (Mercier et al., 2017). The output of this operation is filtered, to remove power line effect (50 Hz and harmonics in Europe), using a FIR bandstop filter with 2 Hz bandwidth. Then, each SEEG recording is transformed using the CWT. With respect to Equation (8.1), the shift parameter τ takes discrete values in $[1, T]$, with T number of points of each time series. The scaling parameter s is a list of 100 elements equally spaced in the logarithmic scale in the interval $[0.3, 3]$. Fixing s, the central frequency f_a of the mother wavelet corresponds to $f_a = (s \cdot t_s)^{-1}$ with $t_s = 1$ ms (sampling period). Consequently, the values of f_a vary in the range between 0.5 Hz and the Nyquist frequency (500 Hz).

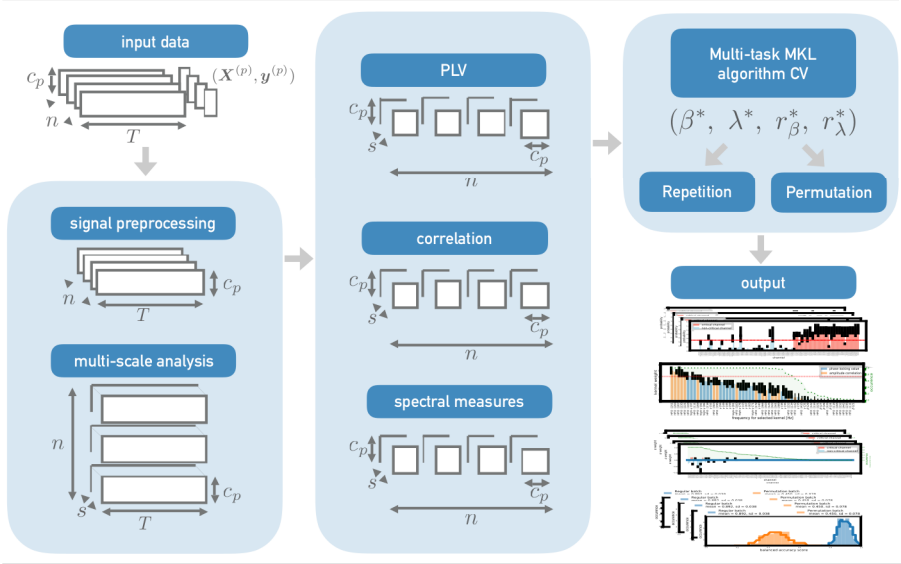


FIGURE 8.2. Schematic representation of the learning pipeline. From top left, SEEG recordings are filtered and converted to a 2D representation using CWT. The central panel represent the similarity measure computation step, applied for each scale of the wavelet transform. In the last panel, the MT-MKL algorithm learns the optimal hyper-parameters. This final step is repeated to obtain statistics on the parameters $(\boldsymbol{w}, \boldsymbol{\alpha})$, the vector of classification probabilities and permutation test results.

In step *(ii)*, the multi-scale representation is given as input to different similarity measures (PLV, correlation and spectral measures). For each patient p, data are transformed into $k = 3 \times 100$ kernels, each of dimension $c_p \times c_p$ (number of channels for a patient p). The computation of spectral measures over all the time series is heavily intensive. Hence, this quantity was approximated by averaging its estimation on smaller, non-overlapping windows of the signal (5.9 seconds length each).

Finally, in step *(iii)*, the MT-MKL is applied on the resulting kernels. In particular the data set is split, for each patient, in half channels for the learning set and the other half for the validation set. The proportion between critical and non-critical channels in each set is kept fixed. The learning set is used to select optimal hyper-parameters with a MCCV procedure over a grid of parameters, and the score is computed on the validation set. The procedure is repeated 50 times to assess the stability of the result. The best hyper-parameters were selected based on the average balanced accuracy over all patients. The outcomes of the pipeline are: *(a)* a vector $\boldsymbol{w}$ which weights the similarity measures, shared across all patients, *(b)* measures on single subject that include statistics on the set of coefficients $\boldsymbol{\alpha}$, to classify previously unseen channels, *(c)* statistics on the probability of each channel of being critical, and *(d)* scores for the classification task and permutation test.

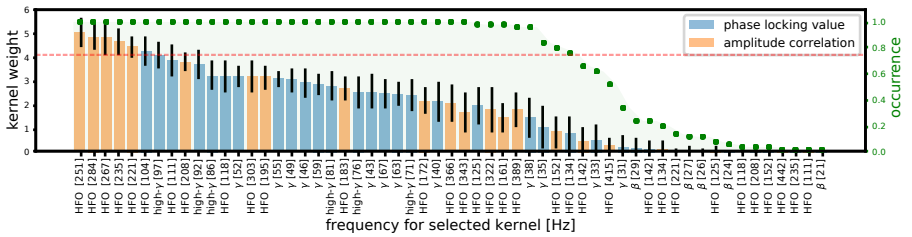


FIGURE 8.3. Kernels contributing to the characterisation of the epileptogenic areas, indicated with the central frequency of the mother wavelet and the event type related to each frequency. Each bar corresponds to weight average and standard deviation through the repetitions of the experiment. The right y-axis denotes the occurrence, the green dots correspond to the number of times each kernel was selected throughout the repetitions. The dashed line indicates the 0.75% occurrence value.

8.2.6 Results

The learning pipeline has different outputs starting from the neural recordings of multiple patients. This section focuses on the selection of the relevant channels for the discrimination task, which will be used in the second part of this chapter. Instead, (D'Amario et al., 2018) contains a complete set of results on the data analysis pipeline as introduced in this section.

Figure 8.3 shows the kernels which were selected at least once, ordered by their total occurrence throughout the 50 repetitions of the experiment. The imposition of the $\ell_1\ell_2$ penalty in Equation (8.3) results in a sparse, stable and small-normed vector $\boldsymbol{w}$. The non-zero components of this vector indicate the importance of similarity measures at specific frequency bands for the prediction task. Indeed, such result highlights the contribution of high frequency events to signal classification.

The highest components of $\boldsymbol{w}$ correspond to amplitude correlation at high frequency and phase synchrony at γ and high-γ bands. Without imposition of any prior knowledge, the method confirms the relevance of high frequencies and abrupt changes in the brain activity for the localisation of pathological areas in focal epilepsy. The selection of ripples and fast ripples events confirms the recent result of Fedele et al. (2017), which shows that the co-occurrence of the two patterns allows for a more precise localisation of the EZ. Also, co-occurrence of γ and high-γ bands with HFO events are confirmed to play a relevant role in the definition of EZ, as observed by Diessen et al. (2013).

This pipeline allows to highlight both important similarity measures for the classification task (across patients) and relevant channels for each patient. Figure 8.4 shows the channels importance for the prediction task of a particular patient. Recalling Equation (8.3), the classification coefficients $\boldsymbol{\alpha}^{(p)}$ for each patient p are estimated based on repetitions of the experiment. Each component of $\boldsymbol{\alpha}^{(p)}$ corresponds to a specific channel selected during the training phase. This vector is constrained to be sparse and small normed via the $\ell_1\ell_2$ regularisation term. Figure 8.4 shows the highest values of $\boldsymbol{\alpha}^{(p)}$ across

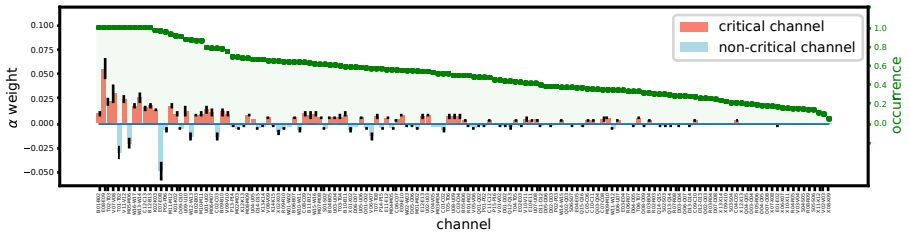


FIGURE 8.4. Channel importance for the prediction for a single patient.

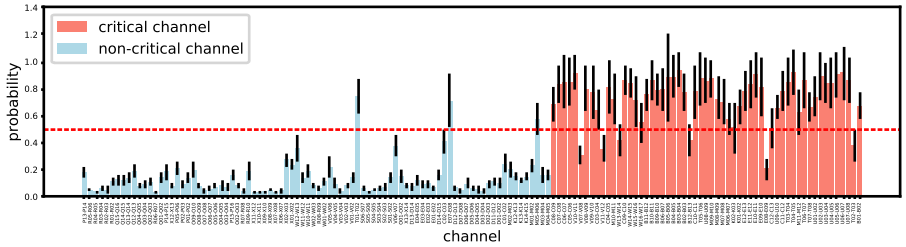


FIGURE 8.5. Probabilities of each channel to be critical.

repetitions. Each channel is related to its occurrence as the number of times its coefficient was non-zero on the number of times it was selected during the training phase. In the plot, channels are ordered both in their average coefficient value and occurrence. Note that the algorithm correctly assigns positive weights to critical channels and negative weights to non-critical ones in all cases.

Figure 8.5 shows the probabilities of each channel to be critical, compared to true critical and non-critical channels (red/blue) as tagged by clinical experts. The logistic function quantifies the probability of belonging to the class of critical channels, hence provides, for each patient, the statistics on probability values for the critical class when selected as test samples. The red dashed line corresponds to 50% probability. This allows to individuate the channels corresponding to less reliable prediction. Note that generally most of the channels are far from chance.

Such learning pipeline favoured the identification of interesting channels where to further investigate for their interactions over time, a process described in what follows.

8.3 *Wishart Process*

The idea of Wishart processes is closely related to Gaussian processes (GPs), since the GWP is built starting from GPs. A Gaussian process is a collection of random variables, for which (any finite number) have a joint Gaussian distribution (Rasmussen, 2004). Hence, it is possible to define a distribution over functions $u(z) \sim \mathcal{GP}(m(z), \kappa(z, z'))$, where z is an arbitrary dependent

variable, and the mean $m(z)$ and kernel function $\kappa(z, z')$ are defined as follows:

$$m(z) = \mathbb{E}[u(z)], \tag{8.4}$$

$$\kappa(z, z') = \text{cov}[u(z), u(z')]. \tag{8.5}$$

Any collection of function values has a joint Gaussian distribution, as follows:

$$(u(z_1), \ldots, u(z_N))^\top \sim \mathcal{N}(\boldsymbol{\mu}, K), \tag{8.6}$$

where $K \in \mathcal{S}_+^N$ is the Gram matrix which encodes the kernel between the dependent variables z, *i.e.*, $K_{ij} = \kappa(z_i, z_j)$, and the mean $\boldsymbol{\mu} = (m(z_i))_{i \in \{1,\ldots,N\}}$. The choice of a particular kernel determines the property of such functions (*e.g.*, smoothness, periodicity).

Similar to constructing a Wishart distribution starting from multivariate Gaussian distribution, the idea is to build the generalised Wishart process starting from Gaussian processes (Wilson and Ghahramani, 2011). In particular, this chapter assumes the dependent variable z to be the time indexing, even if the construction allows z to assume values from any arbitrary set. For simplicity of the notation, temporal indexing is indicated with t.

Consider vd independent Gaussian process functions, such that $u_{ij}(t) \sim \mathcal{GP}(0, \kappa)$ with $i = 1, \ldots, v$ and $j = 1, \ldots, d$. Particularly, $\text{cov}[u_{ij}(t), u_{ij}(t')] = \kappa(t, t')\delta_{ii'}\delta_{jj'}$ and $(u_{ij}(t_1), \ldots, u_{ij}(t_N))^\top \sim \mathcal{N}(0, K)$, where δ is the Kronecker delta, and $K \in \mathbb{R}^{N \times N}$ is the kernel matrix with elements $K_{ij} = \kappa(t_i, t_j)$.

Following the notation of Wilson and Ghahramani (2011), let $\hat{\boldsymbol{u}}_i(t) = (u_{i1}(t), \ldots, u_{id}(t))^\top$, and L be the Cholesky decomposition of a scale matrix $V \in \mathcal{S}_{++}^d$, such that $LL^\top = V$. In this setting, the covariance matrix $\Sigma(t)$ at each time point t has a Wishart marginal distribution (Section 2.2):

$$\Sigma(t) = \sum_{i=1}^{v} L\hat{\boldsymbol{u}}_i(t)\hat{\boldsymbol{u}}_i^\top(t)L^\top \sim \mathcal{W}_d(v, V), \tag{8.7}$$

under the constraint that the kernel function $\kappa(t, t) = 1$. Each element $\hat{\boldsymbol{u}}_i(t)$ is a univariate Gaussian with zero mean and variance $\kappa(t, t) = 1$. Given the fact that each one of this elements are uncorrelated, $\hat{\boldsymbol{u}}_i(t) \sim \mathcal{N}(\mathbf{0}, I)$, and $\mathbb{E}[L\hat{\boldsymbol{u}}_i(t)\hat{\boldsymbol{u}}_i^\top(t)L^\top] = LIL^\top = LL^\top = V$, which leads to $L\hat{\boldsymbol{u}}_i(t) \sim \mathcal{N}(0, V)$. Since Equation (8.7) includes a sum across v outer products of $\mathcal{N}(0, V)$ random variables, according to Definition 2.1, the covariance at a time t follows a Wishart distribution $\mathcal{W}_d(v, V)$. Let $\Sigma(t) \sim \mathcal{GWP}(V, v, \kappa(t, t'))$ denote that $\Sigma(t)$ is a sequence of positive semi-definite random matrices with $\mathcal{W}_d(v, V)$ marginals. Hence, a draw from the Wishart process is a collection of matrices indexed by time, like a draw from a Gaussian process is a collection of function values indexed by time. Figure 8.6 shows an example for a two-dimensional Wishart process.

8.3.1 Inference

Consider a data set D, composed of the observations $(\boldsymbol{x}^i(t))_{1 \leq i \leq n_t}$, $\boldsymbol{x}^i(t) \in \mathbb{R}^d$, $t = 1, \ldots, T$. Assume to have a generalised Wishart process prior on the

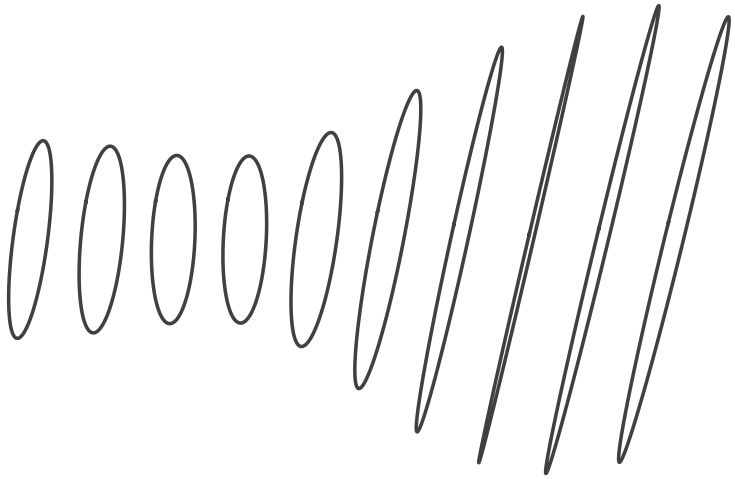


FIGURE 8.6. A draw from the Wishart process. Each ellipse represents a 2-dimensional covariance matrix indexed by time (from left to right). The ellipse representation of a covariance matrix is given by the correlation between the variables (rotation), and the eigenvalues of the matrix (major and minor axes).

sequence of covariance matrices in time, *i.e.*, $\Sigma(t) \sim \mathcal{GWP}(V, \nu, \kappa)$. A Bayesian inference procedure allows to sample from the posterior distributions given the observations using the Gibbs sampling (Geman and Geman, 1984). After the initialisation of $\boldsymbol{u}, \boldsymbol{\theta}, L, \nu$, the sampling algorithm consists in the following iterative steps:

$$p(\boldsymbol{u}|\boldsymbol{\theta}, L, \nu, D) \propto p(D|\boldsymbol{u}, L, \nu)p(\boldsymbol{u}|\boldsymbol{\theta}) \tag{8.8}$$

$$p(\boldsymbol{\theta}|\boldsymbol{u}, L, \nu, D) \propto p(\boldsymbol{u}|\boldsymbol{\theta})p(\boldsymbol{\theta}) \tag{8.9}$$

$$p(L|\boldsymbol{\theta}, \boldsymbol{u}, \nu, D) \propto p(D|\boldsymbol{u}, L, \nu)p(L) \tag{8.10}$$

$$p(\nu|\boldsymbol{\theta}, \boldsymbol{u}, L, D) \propto p(D|\boldsymbol{u}, L, \nu)p(\nu) \tag{8.11}$$

which will converge to samples from $p(\boldsymbol{u}, \boldsymbol{\theta}, L, \nu|D)$.

As before, $\boldsymbol{u}$ are the Gaussian process functions. In particular, $\boldsymbol{u}$ is a vector of length Ndv. The prior $p(\boldsymbol{u}|\boldsymbol{\theta}) = \mathcal{N}(0, K_B)$ is a Gaussian distribution where K_B is a block diagonal covariance matrix formed using dv of the K matrices. If the K matrices depend from dimensions d or degree of freedom ν, then such matrices will be different from one another. Sampling from (8.8) exploits the elliptical slice sampling (Murray, Prescott Adams, and MacKay, 2010), which jointly updates every element of $\boldsymbol{u}$, and it was designed to sample from posteriors with correlated Gaussian priors (Wilson and Ghahramani, 2011).

The prior on the parameters depends on the data. In general, Wilson and Ghahramani (2011) suggest to sample from (8.9) and (8.10) using the Metropolis-Hastings algorithm (Hastings, 1970), with a lognormal prior on θ and a spherical

TABLE 8.1. Subset of the interesting channels. Almost all of them reside in different regions of the brain. Three of them are tagged as epileptogenic (1) by the clinical expert, while the others are tagged as not epileptogenic (-1).

channel	tag	region	channel	tag	region
B01-B02	1	Hip	R01-R02	-1	Wm-sINS
V08-V09	1	Wm	O02-O03	-1	Wm
C03-C04	1	meOTS	X01-X02	-1	CIa

normal prior on L. Instead of learning ν, also, the authors show to be effective to set $\nu = d + 1$.

8.4 Wishart Process for Epilepsy Data

The sampling algorithm (Section 8.3.1) was applied on the epilepsy data used in Section 8.2. Here, consider N data points, one at each time t. The dimensions d corresponds to the channels.

The following analysis involves a single patient, for which only channels included in Table 8.1 were analysed due to computational restrictions. Indeed, such channels were selected as relevant based on Figures 8.4 and 8.5, and their belonging to different part of the brain. Also, the time-series was restricted to 1000 ms (starting from 504000 ms, to avoid border effects). Figure 8.8 shows the results of the Wishart process for such channels, including a covariance matrix averaged for each 20 ms.

The general correlation between channels is in line with their characteristics (tag and region). For example, V08-V09 and O02-O03 exhibit a certain pattern of correlation in the first part of the series, decreasing (to almost zero) in the second part. While their tag is different (V08-V09 is tagged as epileptogenic, O02-O03 is not), they belong to the same area of the brain (white matter). C03-C04 and X01-X02, C03-C04 and O02-O03, V08-V09 and R01-R02 exhibit almost zero correlation. This is reasonable given the information on their tag and the region of the brain associated, which is different among them. Likewise, the temporal correlation between B01-B02 and C03-C04 remains low. While the tag associated to such channels is the same (both epileptogenic), they belong to different part of the brain (Hippocampus and meOTS).

Also, R01-R02 and X01-X02 exhibits a low amount of (positive) correlation. Indeed, while belonging to different areas of the brain, both channels were tagged as epileptogenic by the medical experts.

Channels B01-B02 and O02-O03 belong to different part of the brain and are marked with different tags. However, their correlation is (in some time points) higher than 0. Figure 8.7 shows a zoomed portion of the time series for visual inspection, where the covariance between channels B01-B02 and O02-O03 varies across time. Notably, the covariance between the two channels is generally higher than zero, so that there is a positive correlation between B01-B02 and O02-O03. This is not true for each point of the series, which

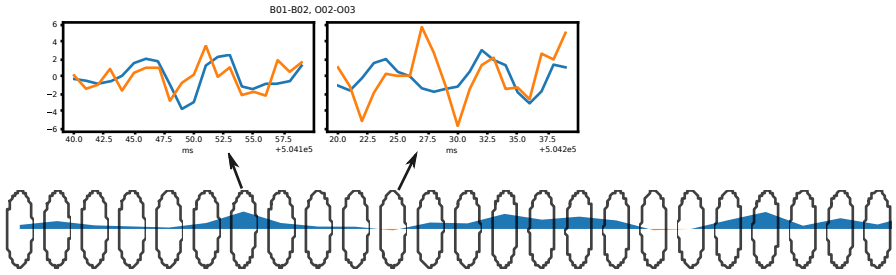


FIGURE 8.7. Visual inspection on the covariance matrix for B01-B02 and O02-O03 channels. Each covariance matrix is averaged in 20 ms. For two covariance matrix in the particular time-points, we plot the time-series associated.

shows an oscillating behaviour. Indeed, the time-series associated to such covariance matrices exhibit an discernible correlation when the covariance is higher than zero, while they appear to be independent when the covariance is zero.

8.5 Discussion

Epilepsy is a disorder which cause severe impairments in daily life. More than 50 million people are affected by such disease and many needs surgical operations. The identification of a minimal amount of neural cortex to ablate from the patient is a crucial and challenging task, often based on invasive procedures such as SEEG. Nonetheless, characterisation of SEEG signals is burdensome and time-consuming, usually based on signal processing tools or visual inspection.

This chapter proposes a novel data analysis pipeline on SEEG signals, in order to automatic classify the neural recordings and quantitatively assess their inter-correlation, allowing for in-depth characterisation of the SEEG signals. The pipeline allows the pairwise comparison between multi-scale representation of the time series, in such a way to automatically select the most relevant similarity measures at specific frequency bands and to differentiate pathological activity from physiological in focal epilepsy. The learning pipeline was applied to a data set of 18 patients for a total of 2347 neural recordings analysed by medical experts. Without any prior assumption on the problem, the data-driven method revealed the most discriminative frequency bands for the localisation of epileptic areas in the high-frequency spectrum ($\geq$ 80 Hz) while showing high performance metric scores, which represents a starting point for the search of clinical biomarkers of epileptogenicity.

Temporal characterisation of the channels using Wishart processes offers insight on the developing relations across different brain areas, considering each time point as a different sample which belongs to a similar but distinct distribution, with respect to the previous and subsequent time point. Furthermore, appropriate kernel functions for the evolutionary behaviour of the time series would favour additional characterisation of SEEG signals at high frequencies

($\geq$ 80 Hz), highlighting high frequency patterns which are informative for the critical/non-critical state discrimination.

To the best of the author's knowledge, this represents a first attempt to integrate multi-scale kernel representation of neural signals for EZ localisation, in a context of multi-task classification and kernels selection. The proposed pipeline shows optimal performance and provides a starting point in the direction of data-driven definition of clinical biomarkers and of a general deeper understanding of focal epilepsy.

This chapter focuses on the interictal phase. Preictal and ictal phases may offer further insights, since the synchronisation level changes across these stages (Burns et al., 2014). With the help of Wishart processes for the characterisation of evolution patterns of the time series, preictal and ictal phases may help the identification of different frequency bands and the classification outcome.

These steps represent further efforts in the direction of a personalised medicine approach to focal epileptic patients, with the double aim of explicitly understand the main features of the pathology and detect the EZ in the most efficient and precise way possible.

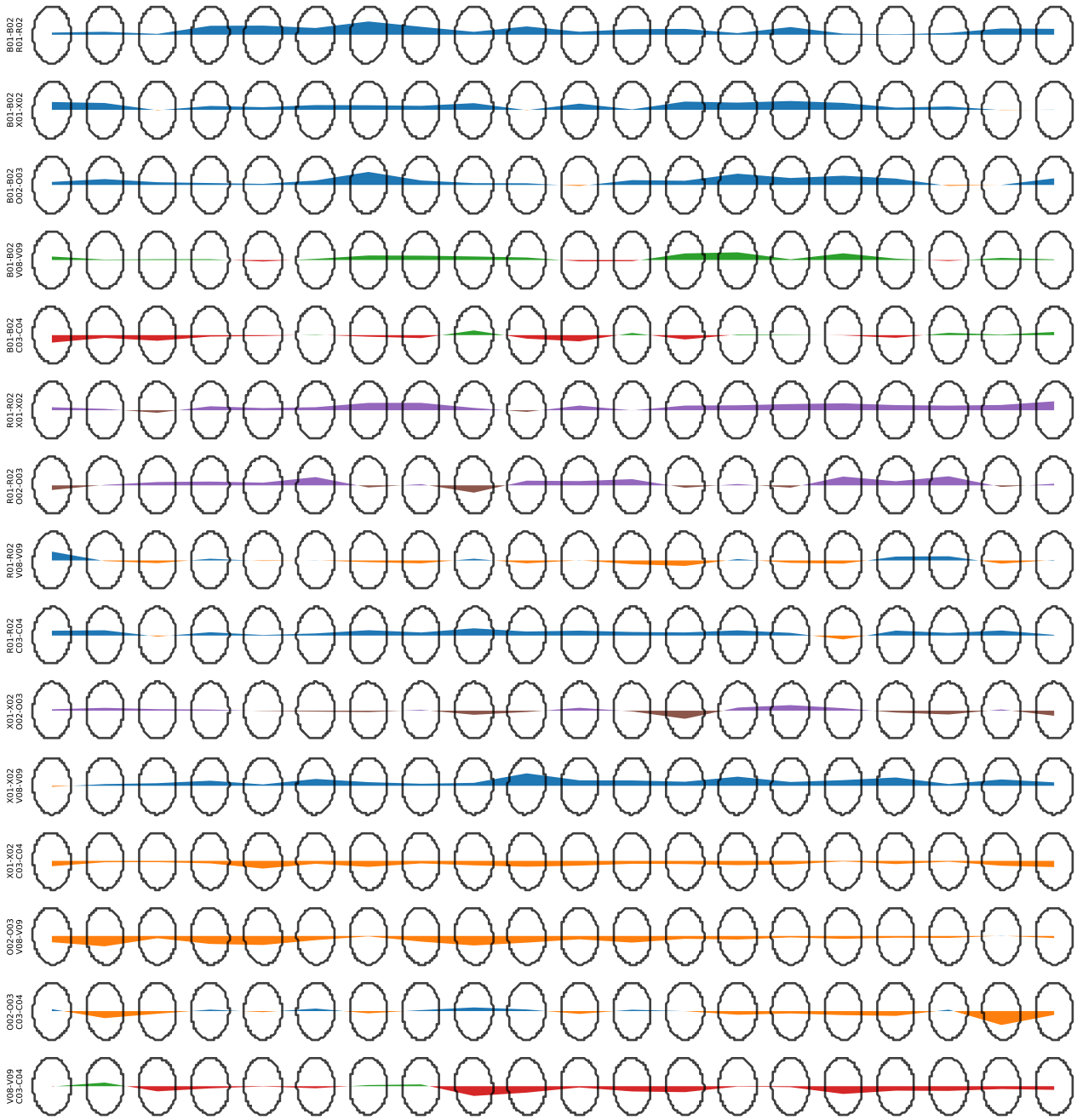


FIGURE 8.8. Wishart process for the channels as detailed in Table 8.1. The covariance matrices belonging to channels with the same tag have a correlation coefficient in green/red for positive tags, purple/brown for negative tags, while blue/orange indicates a correlation between channels with opposite tag.

Conclusion and Future Work

The work of this thesis focuses on the graphical modelling of multivariate time series, and aims at extending recent advances in dynamical network inference. In particular, from the methodological point of view this thesis contains two main contributions.

The first contribution leverages on recent advances in optimisation theory resulting in a novel forward-backward splitting (FBS) procedure for the time-varying graphical lasso model, allowing to scale and speed-up the computation in some common use cases (Chapter 4). Also, the FBS procedure avoids variable duplication, which is fundamental to analyse a large set of variables in real contexts.

The second main contribution consists in the development of a novel graphical modelling method, that is the latent variable time-varying graphical lasso, to model a dynamical system while taking into account latent factors that may change during the evolution of the system (Chapter 5 and Tomasi et al., 2018b). Such contributions have been extensively validated on both synthetic and real data, showing empirical advantages in using such models over the state-of-the-art methods for graphical modelling.

Graphical models for temporal (or pseudotemporal) data have been the centre of the work of this thesis also from an application point of view, in particular for breast cancer progression (Chapter 6), haematopoietic stem cells (Chapter 7) and epilepsy data (Chapter 8). These works are suitable for the application of the developed dynamical graphical modelling methods.

Both proposed methods in Chapters 4 and 5 may be further improved, such as exploiting the structure of the involved matrices (*e.g.*, the block structure of precision matrices) to increase the efficiency of the implemented optimisation algorithms, such as the computation of ∇f in Equation (4.1). In particular, consider the latent variable time-varying graphical lasso. Here, the L matrix is natively not decomposable, because the information on the latent factors, as well as the information on the interactions between latent and observed variables. Additional information on, for example, the matrix Θ_{OH} as in Equation (2.16), allows to decompose the L matrix into its three components. Furthermore, matrix factorisation methods may in general head to the inference of the exact contribution of latent factors starting from the $\boldsymbol{L}$ matrices (Ding, He, and Simon, 2005; Tozzo et al., 2018).

Also, a notable remark involves the relevant similarity between latent variable and data integration methods. Interpreting Equation (2.16) as two different networks (one between variables H and one between variables O), the idea of network integration may be to use the information between the variables H and O to estimate both networks between H and between O (Cheng, Shan, and Kim, 2017). Starting from this, the latent variable time-varying graphical lasso

method may be adapted for a data integration method with the information on the temporal evolution of the data.

Future improvements may also involve the minimisation of $\boldsymbol{\Theta}$ in the time-varying graphical lasso algorithm under FBS minimisation. A bottleneck of the method presented in Section 4.1 involves the need of the search for γ to ensure the positive-definiteness of the matrix under minimisation. However, consider the lower Cholesky decomposition of the precision matrix $\Theta_t = C_t C_t^\top$. Such decomposition and the following minimisation of C allow to ensure the positive-definiteness of the matrix Θ_t implicitly. In this context, minimising the C matrix would be sufficient, so to avoid the γ step in Algorithm 3 which only ensure the positive-definiteness of Θ_t.

Furthermore, the temporal penalty which both the time-varying graphical lasso and latent variable time-varying graphical lasso exploit only involves consecutive time points. Instead, it would be possible to use different temporal kernels to model diverse behaviours of variable interactions, such as periodicity (*e.g.*, in circadian cycles), an approach similar to Wishart processes (Wilson and Ghahramani, 2011).

The graphical modelling of time-series is a relevant topic in machine learning. As such, this thesis represents a starting point in the direction of developing appropriate models for time series and data analysis, aiming at a better understanding of underlying processes to improve pattern recognition tasks.

Availability and Implementation

The code developed for the work included in this thesis is freely available online. The minimisation algorithms of time-varying graphical lasso with forward-backward splitting (Chapter 4), latent variable time-varying graphical lasso (Chapter 5), the group lasso with overlap, Gaussian discriminant analysis (Chapter 6) and the Wishart process inference (Chapter 8) are included into REGAIN, a open-source Python library, available under BSD-3-Clause at `https://github.com/fdtomasi/regain`. REGAIN is fully compatible with the SCIKIT-LEARN library of machine learning algorithms, providing a straightforward and intuitive interface. The implementation relies on low-level high-performance libraries for numerical computations and it exploits closed-form solutions for proximal operators, leading to a fast and scalable optimisation algorithm even with an increasing number of unknowns.

The multi-task multiple kernel learning pipeline (Section 8.2) is available online as an open-source Python framework, available under BSD-3-Clause at `https://github.com/fdtomasi/multikernel`. The implementation relies on high-performance libraries for numerical computations, scaling properly to an arbitrary number of patients and acquisition areas per patient.

Part IV

Appendix

This appendix contains further details on the methods presented in this thesis. In particular, Appendix A contains useful mathematical notions for the theory used in Chapters 4 and 5. Appendix B includes additional work developed in a parallel direction with respect to the core of this thesis. Such work is contained in (Tomasi et al., 2019).

A Linear Algebra

This appendix presents useful theorems and properties that have been used throughout this thesis. Other details (and proofs) of such results may be found in (Harville, 1997; Lauritzen, 1996; Murphy, 2012).

Appendix A.5 shows omitted steps for the derivation of ADMM that have been used in Section 5.2, namely the minimisation equivalence in the derivation of ADMM.

A.1 Graph Theory

Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of vertices of the graph and $\mathcal{E}$ is the set of edges. The boundary $\text{bd}(A)$ of a subset A of vertices is the set of vertices in $\mathcal{V} \setminus A$ that are neighbours to vertices in A. The closure $\text{cl}(A)$ of A is defined as $A \cup \text{bd}(A)$.

Definition A.1 (Decomposition of graphs)**.** *A pair (A, B) of subsets of the vertex set $\mathcal{V}$ of an undirected graph $\mathcal{G}$ is said to form a decomposition of $\mathcal{G}$ if $\mathcal{V} = A \cup B$, $A \cap B$ is complete and separates A from B.*

In this case, (A, B) decomposes the graph $\mathcal{G}$ into the components $\mathcal{G}_A$ and $\mathcal{G}_B$. A decomposable graph is a graph that can be decomposed into its cliques, that is, formally, follows the next definition.

Definition A.2 (Graph decomposability)**.** *An undirected graph is said to be decomposable if it is complete, or if there exists a proper decomposition (A, B) into decomposable subgraphs $\mathcal{G}_A$ and $\mathcal{G}_B$.*

A.2 Matrix Results

Definition A.3 (Inverse of a partitioned matrix)**.** *The inverse of a partitioned matrix is given by*

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} = \begin{pmatrix} E^{-1} & -E^{-1}G \\ -FE^{-1} & D^{-1} + FE^{-1}G \end{pmatrix} \tag{A.1}$$

where $E = A - BD^{-1}C$, $F = D^{-1}C$, and $G = BD^{-1}$.

A.3 Trace

Definition A.4 (Trace)**.** *The trace of a matrix $\boldsymbol{X}$ is denoted with $\text{tr}(\boldsymbol{X})$, and it is defined as the sum of its diagonal:*

$$\text{tr}(\boldsymbol{X}) = \sum_i X_{ii}. \tag{A.2}$$

Property A.1 (Inner product). *If $\langle \cdot, \cdot \rangle$ is the standard inner product on $\mathbb{R}^n$, then*

$$\operatorname{tr}(A^\top B) = \langle A, B \rangle. \tag{A.3}$$

Property A.2 (Cyclic permutation)**.**

$$\operatorname{tr}(ABC) = \operatorname{tr}(CAB) = \operatorname{tr}(BCA) \tag{A.4}$$

A.4 Derivatives

Proposition A.1 (Trace)**.**

$$\frac{\partial \operatorname{tr}(BA)}{\partial A} = B^\top \tag{A.5}$$

Proposition A.2 (Determinant)**.**

$$\frac{\partial \det(A)}{\partial A} = \operatorname{adj}^\top(A) \tag{A.6}$$

Proposition A.3 (Logarithm of the determinant)**.**

$$\frac{\partial \operatorname{logdet}(A)}{\partial A} = \frac{\operatorname{adj}^\top(A)}{\det(A)} = A^{-\top} \triangleq (A^{-1})^\top \tag{A.7}$$

A.5 Minimisation Equivalence

Consider the following minimisation problem:

$$\arg\min_{\Theta} f(\Theta) + \frac{\rho}{2}\|\Theta - A\|^2 + \frac{\rho}{2}\|\Theta - B\|^2 + \frac{\rho}{2}\|\Theta - C\|^2 \tag{A.8}$$

A minimiser is given by:

$$0 \in \partial f(\Theta) + \rho(\Theta - A) + \rho(\Theta - B) + \rho(\Theta - C) \tag{A.9}$$

$$= \partial f(\Theta) + 3\rho(\Theta - \frac{A + B + C}{3}) \tag{A.10}$$

which is the minimiser of:

$$\arg\min_{\Theta} f(\Theta) + \frac{3\rho}{2}\|\Theta - \frac{A + B + C}{3}\|^2. \tag{A.11}$$

Equivalently, this can be shown as follows:

$$\begin{aligned}
2\|\Theta - \frac{A+B}{2}\|^2 &\approx \|\Theta - A\|^2 + \|\Theta - B\|^2 \\
2\|\frac{\Theta - A}{2} + \frac{\Theta - B}{2}\|^2 &= \frac{1}{2}\|\Theta - A\|^2 + \frac{1}{2}\|\Theta - B\|^2 + \langle \Theta - A, \Theta - B \rangle \\
&= \|\Theta - A\|^2 + \|\Theta - B\|^2 + \langle \Theta - A, \Theta - B \rangle \\
&\quad - \frac{1}{2}\|\Theta - A\|^2 - \frac{1}{2}\|\Theta - B\|^2 \\
&= \|\Theta - A\|^2 + \|\Theta - B\|^2 - 2\|\frac{\Theta - A}{2} - \frac{\Theta - B}{2}\|^2 \\
&= \|\Theta - A\|^2 + \|\Theta - B\|^2 - \frac{1}{2}\|B - A\|^2,
\end{aligned}$$

hence $\|\Theta - A\|^2 + \|\Theta - B\|^2 = 2\|\Theta - \frac{A+B}{2}\|^2 + \frac{1}{2}\|B - A\|^2$.

B Immunoglobulin Analysis

This chapter introduces an additional work developed over the course of my Ph.D. studies, which is not related to graphical models or time series data, but introduces challenging questions from a biomedical point of view, that is the clonotype identification of antibodies.

Immunoglobulin (IG) clonotype identification is a fundamental open question in modern immunology. An accurate description of the IG repertoire is crucial to understand the variety within the immune system of an individual, potentially shedding light on the pathogenetic process. Intrinsic IG heterogeneity makes clonotype inference an extremely challenging task, both from a computational and a biological point of view. This chapter presents ICING, a framework that allows to reconstruct clonal families also in case of highly mutated sequences. ICING has a modular structure, and it is designed to be used with large next generation sequencing (NGS) data sets, a technology which allows the characterisation of large-scale IG repertoires. The framework is extensively validated with clustering performance metrics on the results in a simulated case. ICING is implemented in Python, and it is publicly available under FreeBSD licence at https://github.com/slipguru/icing.

B.1 Scientific Background

The identification of immunoglobulin (IG) clonotypes is a key question in modern immunology. A clonotype is a particular combination of IGs generated by a single plasma cell clone, which is a population of cells all derived from a single progenitor cell (germline). The ability to infer clonotypes is crucial as it allows to understand how much diversity an individual has in its immune repertoire and to study immune response through B-cell clonal amplification and diversification. Indeed, understanding the variety within the immune system of an individual may potentially shed light on pathogenetic processes. In healthy individuals the repertoire is expected to be extremely diverse, to guarantee the ability to respond to a wide range of antigens (*e.g.* bacteria, viruses). The diversity of the B-cell repertoire is due to the gene recombination process, where, by random selection, one for each V, D and J genes are joined together, with a simultaneous trimming and addition of random nucleotides (Figure B.1). The resulting bridging segment between V and J genes, called complementarity determining region 3 (CDR3), is the most variable and therefore important for the antigen binding (Rock, 1994). Before encountering an antigen, B-cells have zero (or few) somatic mutations. Without considering mutations, the overall repertoire diversity usually comprises 10^7 to 10^8 clonotypes, with lower bounds of diversity of 10^5 and potentially as high as 10^{11} unique molecules in a single individual (Glanville, 2009). After the

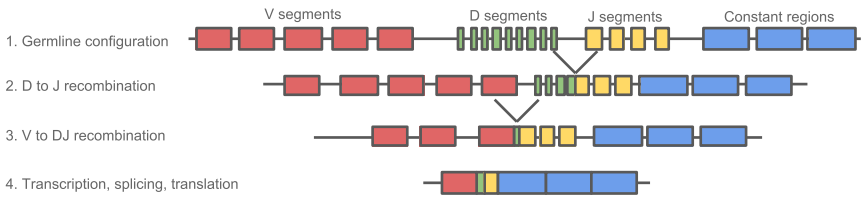


FIGURE B.1. IG recombination. Starting from V(D)J gene segments, one of each type is selected to produce the IG sequence. When joining two segments, some insertions and deletions (*indels*) may occur. A constant region is appended to the IG sequence after the recombination.

immune response, they undergo clonal amplification and somatic hypermutation, to increase the binding affinity to the antigen (Kleinstein, Louzoun, and Shlomchik, 2003). The potential frequency of somatic hypermutation, which can be at least 10^5-10^6 fold greater than the normal rate of mutation across the genome (Oprea, 1999), may generate many orders of magnitude more diversity in the B-cell receptor repertoire than the 10^{11} unique molecules per individual. Therefore, intrinsic data heterogeneity makes IG clonotyping an extremely difficult task.

B.2 ICING

To tackle the problem of IG clonotyping inference, I developed ICING (Inferring Clonotypes of ImmuNoGlobulins), a Python library publicly available at `https://github.com/slipguru/icing`. The method aims at grouping IGs into clonal families, whose members derive from the same germline ancestor. Input and output data have the same format used by the Change-O suite, hence ICING is easily integrable in the usual pRESTO/Change-O pipeline (Gupta, 2015; Vander Heiden, 2014). In particular, data should be in the format produced by Change-O, that is, IGs should be represented via their V gene calls and CDR3 amminoacidic (or nucleotidic) sequence. Also, an indication of the mutation level of the sequence with respect to reference should be present, to allow for the final steps of the pipeline (Appendix B.3.3).

ICING is designed to be used with a large number of data, for example coming from NGS technologies. The method is implemented in Python, exploiting separate processes on multi-core machines for almost each step of three sequential phases: *(i)* data shrinking, *(ii)* high-level grouping and *(iii)* fine-grained clonotype identification (Figure B.2).

B.3 Materials and Methods

B.3.1 Synthetic Data Generation

Synthetic data sets are generated using *partis* (Ralph and Matsen IV, 2016). Data are characterised by an increasing number of IGs and clones, 0.05 frequency

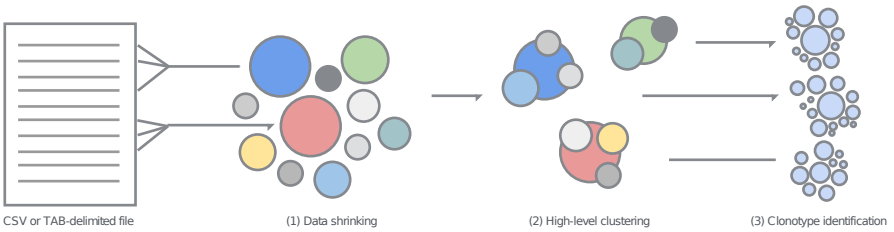


FIGURE B.2. ICING pipeline. Starting from a CSV or TAB-delimited file, the first step consists in grouping together sequences based on their V gene calls and CDR3 identity (data shrinking step). An high-level clustering is done on CDR3 lengths to reduce the computational workload of the third and final phase, which involves a clustering step on each of the previously found groups to obtain fine-grained IG clonotypes.

of insertions and deletions (*indels*) of maximum length of 6 nucleotides on the CDR3 sequence, and different degrees of V gene sequence mutation level. In particular, synthetic data sets contained from 10^4 to 10^6 records, divided into 100 to 3000 clonotypes, respectively. Table B.1 presents an overview of the data sets.

B.3.2 Preprocessing

The data sets were submitted to IMGT/HighV-QUEST (Alamyar, 2012) for V(D)J genes inference, then preprocessed by a Change-O feature (Gupta, 2015). The outcome is a single TAB-delimited file containing the information about IGs and their metadata, such as the identification of V(D)J sequences (*i.e.*, V(D)J gene calls), V gene sequence mutation level and identification of CDR3 sequence, to be used as input to the pipeline.

B.3.3 Clonotype Identification

The clonotype identification step is divided into three parts.

DATA SHRINKING. Input data are grouped based on V gene calls (exact correspondence) and CDR3 identity (completely overlapping sequence). This allows to reduce the computational workload of next clustering steps. To each group is assigned a weight, equal to the cardinality of the group.

HIGH-LEVEL GROUP INFERENCE. This phase involves a clustering step on CDR3 lengths of previously identified groups. The outcome, which consists of high-level groups of IGs to be refined afterwards, contains IG sequences having comparable CDR3 lengths. This is done using MiniBatchKMeans clustering algorithm (Sculley, 2010), which is computationally efficient and, more importantly, may group together very similar clusters.

FINE-GRAINED GROUP INFERENCE. Each high-level group extracted before is then subdivided based on the actual IG distance. The distance between

TABLE B.1. Datasets overview. For reference, the total number of functional gene segments for the V/D/J regions of heavy chains in the human genome are 65/27/6 (Janeway et al., 1997).

data set	sequences	clonotypes	avg seqs/clone	unique V genes	unique D genes	unique J genes	mean (std) of V gene mutation
D1	9233	77	92.35	35	24	6	9.59 (4.64)
D2	17825	74	185.09	38	24	6	8.64 (4.46)
D3	37897	77	396.43	34	25	6	9.04 (4.51)
D4	47764	389	99.08	56	25	6	8.63 (4.30)
D5	102336	388	209.44	58	25	6	8.41 (4.70)
D6	205986	379	428.44	56	25	6	9.56 (4.46)
D7	162713	1168	109.66	58	25	6	8.72 (4.67)
D8	301978	1180	206.22	58	25	6	9.15 (4.73)
D9	589680	1185	400.26	58	25	6	8.94 (4.65)
D10	291076	2282	96.29	58	25	6	8.84 (4.46)
D11	568799	2317	187.76	58	25	6	9.12 (4.76)
D12	1208110	2358	404.30	58	25	6	9.11 (4.77)

IGs is computed taking into account V gene calls and CDR3 sequences. In particular, the distance between two IGs is lower than infinity if and only if they have at least one V gene call in common. In such case, their actual distance is computed using a sequence distance method on their CDR3 sequences. In particular, the method implements a generic normalised distance measure based on a particular model matrix $\mathcal{M}$. Let $\|\mathcal{M}\|_{\max} = \max_{i,j} |\mathcal{M}_{ij}|$. For two sequences s and t of equal length ℓ, their distance $\mathcal{D}(s,t)$ is defined as:

$$\mathcal{D}(s,t) = \frac{1}{\ell \cdot \|\mathcal{M}\|_{\max}} \sum_{i=1}^{\ell} \mathcal{M}(s^i, t^i). \tag{B.1}$$

The choice of a specific model depends on the type of data under analysis. When $\mathcal{M} = \mathcal{H}$, where $\mathcal{H}(x,y) = 0$ if $x = y$ and 1 otherwise, the model assumes the form of a normalised Hamming distance (Hamming, 1950).

Such distance measure allows seamless integration of different nucleotidic and amminoacidic models. ICING includes Hamming and its weighted variants, such as HS1F (Yaari, 2013). The models are defined between sequences of equal length. The method allows also the comparison of sequences with different lengths, by tuning a *tolerance* parameter. In such case, a standard alignment step between two sequences of different lengths may be performed before the computation of their distance, using the Smith-Waterman algorithm for sequence alignment (Smith and Waterman, 1981).

IG sequences are characterised by an high level of mutation. Therefore, a correction function based on V gene sequence mutation level may be used to reduce distances between two IGs if mutated. This procedure encodes the uncertainty of the distance measure when dealing with highly mutated data, allowing for a more robust measure. Note that this is a step which is strongly depends on the data at hand. In my experiments, I corrected the distances between two IGs by multiplying $\mathcal{D}(s,t)$ with ν_{st}, where $\nu_{st} = 1 - \frac{m_s - m_t}{2}$, with m_s and m_t are the mutation levels of the sequences s and t, respectively.

After the design of such distance metric, fine-grained groups (*i.e.*, final clonotypes) are extracted using the DBSCAN clustering algorithm (Ester et al.,

1996), which only require the parameter ϵ for the neighbourhood search of spatial distances. On top of an appropriate index structure, the algorithm can run in $O(n \log n)$ and it only needs linear memory, allowing the analysis of large-scale data.

B.3.4 Performance Assessment

For synthetic data sets the information about IG clonotypes is known, and it is used as ground truth. In order to evaluate clustering performance of the method, I used standard metrics such as homogeneity (HOM), completeness (COM) and V-measure (VSC), mutual information based scores, namely Adjusted Mutual Information (AMI) and Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), and Fowlkes-Mallows score (FMI) (Fowlkes and Mallows, 1983; Hubert and Arabie, 1985; Strehl and Ghosh, 2002; Vinh, Epps, and Bailey, 2009). Such measures are bound by $[0, 1]$, and no assumption is made on the cluster structure. Moreover, AMI, ARI and FMI are adjusted against chance, which is an important feature when evaluating a clustering performance in presence of a large number of clusters. Therefore, random (uniform) label assignments have scores close to 0 for measures normalised against chance.

B.3.5 Computing Architecture

Experiments were performed using a computing machine equipped with two Intel® Xeon® CPUs E5-2630 v3 (2.4 GHz, 8 cores each) and 128 GB of RAM[1].

B.4 Results

B.4.1 Performance Evaluation

I evaluated the method performance on the data sets shown in Table B.1. In particular, Table B.2 shows the clustering scores (Appendix B.3.4) for data sets D1–3, obtained using different ICING configurations. The metric used for CDR3 sequence distance computation is the Hamming metric. The other parameters we investigated involve the neighbourhood selection radius of the DBSCAN clustering algorithm (restricted to 0.2 or 0.6), the tolerance of the difference in CDR3 sequence lengths (0, 3 or up to 6 allowed insertions or deletions), and the optional distance correction based on the V gene segment mutation level. Table B.2 is ordered based on a decreasing FMI score, which, for its properties, it is the most strict of the clustering measures described in Appendix B.3.4. The highest scores (close to 1) for each of the three data sets are associated to similar ICING configurations, in which the neighbourhood selection of the DBSCAN clustering algorithm is restricted to 0.2, the tolerance of the difference in sequence lengths is 0 (*i.e.*, no alignment between CDR3s needed to be done), and sequence distances are corrected based on the V gene

[1]This is not representative of the amount of computational resources required by the method.

TABLE B.2. Comparison of performance metrics between various ICING configuration on synthetic data sets. Columns are: ϵ (the DBSCAN parameter for neighbourhood selection), *tolerance* (tolerance parameter on CDR3 length), *correction* (Y for a correction based on the mutation level of V gene segments, N for no correction), followed by the clustering measures as described in Appendix B.3.4. For each data set, results are ordered by a decreasing FMI, which is the most strict of the measures for its properties.

				no chance normalisation				*chance normalisation*		
data set	ϵ	tolerance	correction	HOM	COM	VSC	NMI	AMI	ARI	FMI
D1	0.2	0	Y	0.91	0.94	0.92	0.92	0.90	0.86	0.87
	0.2	6	Y	0.90	0.94	0.92	0.92	0.89	0.86	0.86
	0.2	3	Y	0.87	0.94	0.90	0.90	0.86	0.76	0.78
	0.2	6	N	0.87	0.94	0.90	0.90	0.86	0.75	0.77
	0.2	0	N	0.86	0.94	0.90	0.90	0.85	0.75	0.77
D2	0.2	0	Y	0.93	0.93	0.93	0.93	0.93	0.90	0.91
	0.2	6	Y	0.93	0.93	0.93	0.93	0.93	0.90	0.91
	0.2	3	Y	0.93	0.93	0.93	0.93	0.93	0.90	0.90
	0.2	3	N	0.92	0.93	0.92	0.92	0.91	0.88	0.88
	0.2	0	N	0.91	0.93	0.92	0.92	0.91	0.87	0.88
D3	0.2	0	Y	0.94	0.93	0.93	0.93	0.92	0.92	0.92
	0.2	3	Y	0.94	0.92	0.93	0.93	0.92	0.92	0.92
	0.2	0	N	0.92	0.93	0.92	0.92	0.91	0.89	0.89
	0.2	6	Y	0.92	0.93	0.93	0.93	0.92	0.88	0.88
	0.2	6	N	0.92	0.93	0.92	0.92	0.91	0.87	0.87

segment mutation level. Particularly for data set D1, the distance correction is shown to be a critical step to reliably identify IG clonotypes, as confirmed by high ARI, AMI and FMI scores (chance-corrected clustering measures). Notably, for D2 and D3 data sets, the correction gives better results when associated to a tolerance parameter of 0 or 6 nucleotides for CDR3 sequences.

The best parameters selected on data sets D1–3 were used to evaluate the results on the remaining data sets of Table B.1. The results presented in Table B.3 show that ICING is capable to achieve high performance, which means a reliable IG sequence clonotyping, even with an increasing number of sequences. Also, the method is stable across data sets with different sizes.

B.4.2 Expected Clonotypes

Figure B.3 shows the number of clonotypes found by ICING compared to the expected clonotypes (*ground truth*). Inferred clonotypes are very close to the ground truth disregarding the size of the data sets. This result, together with the high clustering performance achieved by our method (Table B.2 and Table B.3), makes ICING a reliable framework for IG clonotype identification in real contexts, where real clonotypes are not known.

TABLE B.3. ICING results on synthetic data sets, using the best parameters as selected in Table B.2 (ϵ: 0.2, *tolerance*: 0, *correction*: Y). For each data sets, clustering measures are reported as described in Appendix B.3.4.

		no chance normalisation				*chance normalisation*		
data set	sequences	HOM	COM	VSC	NMI	AMI	ARI	FMI
D4	47764	0.90	0.95	0.93	0.93	0.88	0.79	0.80
D5	102336	0.94	0.95	0.94	0.94	0.93	0.89	0.89
D6	205986	0.94	0.95	0.94	0.94	0.94	0.89	0.89
D7	162713	0.93	0.96	0.94	0.94	0.91	0.84	0.84
D8	301978	0.93	0.95	0.94	0.94	0.92	0.86	0.86
D9	589680	0.93	0.96	0.95	0.95	0.92	0.88	0.87
D10	291076	0.94	0.95	0.95	0.96	0.92	0.87	0.86
D11	568799	0.93	0.95	0.94	0.96	0.91	0.89	0.88
D12	1208110	0.95	0.94	0.95	0.95	0.90	0.88	0.90

B.5 Discussion

Synthetic experiments show ICING to be capable of successfully identifying IG clonotypes, using synthetic data comprising highly mutated sequences, different V(D)J recombination events and *indels* on CDR3 sequences. Due to the intrinsic difficulty of validating the method on real data (where the ground truth is not known), this chapter only includes the results obtained on synthetic data, where the method can be validated in relation to the ground truth.

ICING has a modular structure which allows to combine different features. In particular, the clonotype identification step has the potential to include Hamming or other arbitrary nucleotidic or amminoacidic models to compute sequence distances, arbitrary CDR3 length tolerance or V gene sequence mutation-based correction, which is an original contribution of this framework. ICING is scalable with the number of input sequences, allowing for the analysis of large-scale data sets composed of more than 10^6 sequences, which is a typical use-case when dealing with NGS data. To achieve scalability, ICING is based on a novel methodology which exploits the DBSCAN clustering algorithm, on top of an appropriate index structure. In particular, I was not able to compare such pipeline with plain Change-O which, since it is based on hierarchical clustering, has memory complexity of $O(n^2)$, thus infeasible for large data sets. However, i was able to analyse arbitrarily large data sets by exploiting all of the steps shown in Appendix B.3.3, which turned out to be fundamental in the experiments.

ICING is easily integrable in the usual pRESTO/Change-O pipeline for IG analysis and it is ready to be used in real scenarios. In presence of sequences with low rate of recombination and mutation (*i.e.*, as in the case of non-healthy patients), I expect the data shrinking step (Appendix B.3.3) to be highly beneficial for reducing the complexity of the algorithm, which is proportional to the number of unique CDR3 sequences and V gene calls in the data set.

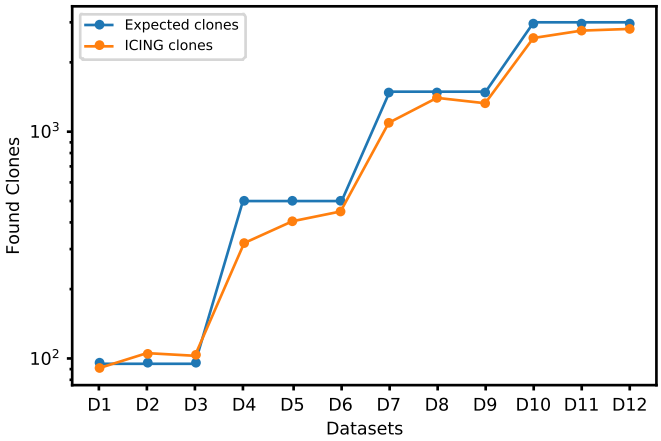


FIGURE B.3. Comparison between ICING clusters and expected clonotypes on synthetic data sets. For each data set (x-axis), the number of clonotypes found by ICING is compared with the expected clonotypes (y-axis), *i.e.*, the *ground truth*. For data sets D1–3, only the best results based on FMI score (Table B.2) are included.

Bibliography

Ahmed, Sumon, Magnus Rattray, and Alexis Boukouvalas (2017). "GrandPrix: Scaling up the Bayesian GPLVM for single-cell data". In: *bioRxiv*, p. 227843 (cited on p. 81).

Akhand, MAH, RN Nandi, SM Amran, and K Murase (2015). "Context likelihood of relatedness with maximal information coefficient for Gene Regulatory Network inference". In: *Computer and Information Technology (ICCIT), 2015 18th International Conference on*. IEEE, pp. 312–316 (cited on p. 7).

Alamyar, Eltaf et al. (2012). "IMGT/HighV-QUEST: the IMGT® web portal for immunoglobulin (IG) or antibody and T cell receptor (TR) analysis from NGS high throughput and deep sequencing." In: *Immunome research* 8.1, p. 26 (cited on p. 110).

Albert, Réka (2007). "Network inference, analysis, and modeling in systems biology". In: *The Plant Cell* 19.11, pp. 3327–3338 (cited on p. 19).

Anandkumar, A., D. Hsu, A. Javanmard, and S. Kakade (2013). "Learning linear Bayesian networks with latent variables". In: *ICML*, pp. 249–257 (cited on p. 46).

Arnulfo, Gabriele, Jonni Hirvonen, Lino Nobili, Satu Palva, and J Matias Palva (2015). "Phase and amplitude correlations in resting-state activity in human stereotactical EEG recordings". In: *Neuroimage* 112, pp. 114–127 (cited on p. 88).

Atay-Kayis, Aliye and Hélène Massam (2005). "A Monte Carlo method for computing the marginal likelihood in nondecomposable Gaussian graphical models". In: *Biometrika* 92.2, pp. 317–335 (cited on p. 11).

Avoli, Massimo, Giuseppe Biagini, and M De Curtis (2006). "Do interictal spikes sustain seizures and epileptogenesis?" In: *Epilepsy currents* 6.6, pp. 203–207 (cited on p. 87).

Bacher, Rhonda and Christina Kendziorski (2016). "Design and computational analysis of single-cell RNA-sequencing experiments". In: *Genome biology* 17.1, p. 63 (cited on p. 80).

Bai, Jushan and Serena Ng (2006). "Evaluating latent and observed factors in macroeconomics and finance". In: *Journal of Econometrics* 131.1-2, pp. 507–537 (cited on p. 60).

Banerjee, Onureena, Laurent El Ghaoui, and Alexandre d'Aspremont (2008). "Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data". In: *Journal of Machine learning research* 9.Mar, pp. 485–516 (cited on pp. 9, 15, 17, 25).

Barabasi, Albert-Laszlo and Zoltan N Oltvai (2004). "Network biology: understanding the cell's functional organization". In: *Nature reviews genetics* 5.2, pp. 101–113 (cited on p. 6).

Beck, Amir and Marc Teboulle (2009). "A fast iterative shrinkage-thresholding algorithm for linear inverse problems". In: *SIAM journal on imaging sciences* 2.1, pp. 183–202 (cited on p. 25).

Bergstra, James and Yoshua Bengio (2012). "Random search for hyper-parameter optimization". In: *Journal of Machine Learning Research* 13.Feb, pp. 281–305 (cited on p. 29).

Bianco-Martinez, E, N Rubido, Ch G Antonopoulos, and MS Baptista (2016). "Successful network inference from time-series data using mutual information rate". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 26.4, p. 043102 (cited on p. 46).

Bien, Jacob and Robert J Tibshirani (2011). "Sparse estimation of a covariance matrix". In: *Biometrika* 98.4, pp. 807–820 (cited on pp. 9, 15).

Bishop, Christopher M et al. (2006). *Pattern recognition and machine learning*. Vol. 4. 4. springer New York (cited on p. 8).

Blainey, Paul C and Stephen R Quake (2014). "Dissecting genomic diversity, one cell at a time". In: *Nature methods* 11.1, p. 19 (cited on p. 79).

Bolstad, Andrew, Barry D Van Veen, and Robert Nowak (2011). "Causal network inference via group sparse regularization". In: *IEEE transactions on signal processing* 59.6, pp. 2628–2641 (cited on p. 62).

Bolte, Jérôme, Shoham Sabach, and Marc Teboulle (2014). "Proximal alternating linearized minimization for nonconvex and nonsmooth problems". In: *Mathematical Programming* 146.1-2, pp. 459–494 (cited on pp. 10, 91).

Borgwardt, Karsten M (2011). "Kernel methods in bioinformatics". In: *Handbook of statistical bioinformatics*. Springer, pp. 317–334 (cited on pp. 89, 90).

Boyd, Stephen, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein (2010). "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers". In: *Foundations and Trends® in Machine Learning* 3.1, pp. 1–122. ISSN: 1935-8237. DOI: 10.1561/2200000016. URL: http://www.nowpublishers.com/article/Details/MAL-016 (cited on pp. 24, 35, 47, 49, 54).

Burchell, Joy M, Richard Beatson, Rosalind Graham, Joyce Taylor-Papadimitriou, and Virginia Tajadura-Ortega (2018). "O-linked mucin-type glycosylation in breast cancer". In: *Biochemical Society Transactions*, BST20170483 (cited on p. 73).

Burns, Samuel P, Sabato Santaniello, Robert B Yaffe, Christophe C Jouny, Nathan E Crone, Gregory K Bergey, William S Anderson, and Sridevi V Sarma (2014). "Network dynamics of the brain and influence of the epileptic seizure onset zone". In: *Proceedings of the National Academy of Sciences* 111.49, E5321–E5330 (cited on p. 100).

Butte, Atul J and Isaac S Kohane (2000). "Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements". In: *Pac Symp Biocomput*. Vol. 5. 415, p. 26 (cited on p. 7).

Cardona, Hernán Dario Vargas, Mauricio A Álvarez, and Álvaro A Orozco (2015). "Generalized Wishart processes for interpolation over diffusion tensor

fields". In: *International Symposium on Visual Computing*. Springer, pp. 499–508 (cited on p. 86).

Chandrasekaran, Venkat, Pablo A Parrilo, and Alan S Willsky (2010). "Latent variable graphical model selection via convex optimization". In: *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*. IEEE, pp. 1610–1613 (cited on pp. 21, 22, 46, 48, 57).

Chandrasekaran, Venkat, Sujay Sanghavi, Pablo A Parrilo, and Alan S Willsky (2011). "Rank-sparsity incoherence for matrix decomposition". In: *SIAM Journal on Optimization* 21.2, pp. 572–596 (cited on p. 21).

Cheng, Lulu, Liang Shan, and Inyoung Kim (2017). "Multilevel Gaussian graphical model for multilevel networks". In: *Journal of Statistical Planning and Inference* 190, pp. 1–14 (cited on p. 102).

Choi, Myung Jin, Venkat Chandrasekaran, and Alan S Willsky (2010). "Gaussian multiresolution models: Exploiting sparse Markov and covariance structure". In: *IEEE Transactions on Signal Processing* 58.3, pp. 1012–1024 (cited on pp. 20, 46).

Choi, Myung Jin, Vincent YF Tan, Animashree Anandkumar, and Alan S Willsky (2011). "Learning latent tree graphical models". In: *Journal of Machine Learning Research* 12.May, pp. 1771–1812 (cited on pp. 12, 20, 46).

Chuu, Chih-Pin, Rou-Yu Chen, John L Barkinge, Mark F Ciaccio, and Richard B Jones (2008). "Systems-level analysis of ErbB4 signaling in breast cancer: a laboratory to clinical perspective". In: *Molecular cancer research* 6.6, pp. 885–891 (cited on p. 71).

Combettes, Patrick L and Băng C Vũ (2014). "Variable metric forward–backward splitting with applications to monotone inclusions in duality". In: *Optimization* 63.9, pp. 1289–1318 (cited on p. 91).

Combettes, Patrick L and Valérie R Wajs (2005). "Signal recovery by proximal forward-backward splitting". In: *Multiscale Modeling & Simulation* 4.4, pp. 1168–1200 (cited on pp. 25, 38).

Condat, Laurent (2013). "A direct algorithm for 1-D total variation denoising". In: *IEEE Signal Processing Letters* 20.11, pp. 1054–1057 (cited on p. 38).

Curtis, Marco de and Giuliano Avanzini (2001). "Interictal spikes in focal epileptogenesis". In: *Progress in neurobiology* 63.5, pp. 541–567 (cited on p. 87).

Danaher, Patrick, Pei Wang, and Daniela M. Witten (2014). "The joint graphical lasso for inverse covariance estimation across multiple classes". In: *Journal of the Royal Statistical Society. Series B: Statistical Methodology* 76.2, pp. 373–397. ISSN: 13697412. DOI: 10.1111/rssb.12033. arXiv: 1111.0324 (cited on pp. 18, 19, 44, 47, 51).

Dawid, A Philip and Steffen L Lauritzen (1993). "Hyper Markov laws in the statistical analysis of decomposable graphical models". In: *The Annals of Statistics*, pp. 1272–1317 (cited on p. 10).

De Ieso, Michael L and Andrea J Yool (2018). "Mechanisms of Aquaporin-Facilitated Cancer Invasion and Metastasis". In: *Frontiers in Chemistry* 6 (cited on p. 73).

Dempster, Arthur P (1972). "Covariance selection". In: *Biometrics*, pp. 157–175 (cited on p. 14).

Diessen, Eric van, Judith I Hanemaaijer, Willem M Otte, Rina Zelmann, Julia Jacobs, Floor E Jansen, François Dubeau, Cornelis J Stam, Jean Gotman, and Maeike Zijlmans (2013). "Are high frequency oscillations associated with altered network topology in partial epilepsy?" In: *Neuroimage* 82, pp. 564–573 (cited on p. 94).

Dimitroff, Charles J (2015). "Galectin-binding O-glycosylations as regulators of malignancy". In: *Cancer research* 75.16, pp. 3195–3202 (cited on p. 73).

Ding, Chris, Xiaofeng He, and Horst D Simon (2005). "On the equivalence of nonnegative matrix factorization and spectral clustering". In: *Proceedings of the 2005 SIAM International Conference on Data Mining*. SIAM, pp. 606–610 (cited on p. 102).

Doulatov, Sergei, Linda T Vo, Stephanie S Chou, Peter G Kim, Natasha Arora, Hu Li, Brandon K Hadland, Irwin D Bernstein, James J Collins, Leonard I Zon, et al. (2013). "Induction of multipotential hematopoietic progenitors from human pluripotent stem cells via respecification of lineage-restricted precursors". In: *Cell stem cell* 13.4, pp. 459–470 (cited on p. 80).

Ehmann, Heike, Christian Salzig, Patrick Lang, Eckhard Friauf, and Hans Gerd Nothwang (2008). "Minimal sex differences in gene expression in the rat superior olivary complex". In: *Hearing research* 245.1-2, pp. 65–72 (cited on p. 82).

Ester, Martin, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. (1996). "A density-based algorithm for discovering clusters in large spatial databases with noise." In: *Kdd*. Vol. 96. 34, pp. 226–231 (cited on p. 111).

Everitt, Brian S (1995). *The Cambridge dictionary of statistics in the medical sciences*. Cambridge University Press Cambridge (cited on p. 66).

Fabregat, Antonio, Konstantinos Sidiropoulos, Phani Garapati, Marc Gillespie, Kerstin Hausmann, Robin Haw, Bijay Jassal, Steven Jupe, Florian Korninger, Sheldon McKay, et al. (2016). "The reactome pathway knowledgebase". In: *Nucleic acids research* 44.D1, pp. D481–D487 (cited on p. 66).

Faith, Jeremiah J, Boris Hayete, Joshua T Thaden, Ilaria Mogno, Jamey Wierzbowski, Guillaume Cottarel, Simon Kasif, James J Collins, and Timothy S Gardner (2007). "Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles". In: *PLoS biol* 5.1, e8 (cited on p. 7).

Farasat, Alireza, Alexander Nikolaev, Sargur N Srihari, and Rachael Hageman Blair (2015). "Probabilistic graphical models in modern social network analysis". In: *Social Network Analysis and Mining* 5.1, p. 62 (cited on p. 45).

Faust, Karoline and Jeroen Raes (2016). "CoNet app: inference of biological association networks using Cytoscape". In: *F1000Research* 5 (cited on p. 8).

Fay, Damien, Hamed Haddadi, Andrew Thomason, Andrew W Moore, Richard Mortier, Almerima Jamakovic, Steve Uhlig, and Miguel Rio (2010). "Weighted spectral distribution for internet topology analysis: theory and applications".

In: *IEEE/ACM Transactions on Networking (ToN)* 18.1, pp. 164–176 (cited on p. 70).

Fedele, Tommaso, Sergey Burnos, Ece Boran, Niklaus Krayenbühl, Peter Hilfiker, Thomas Grunwald, and Johannes Sarnthein (2017). "Resection of high frequency oscillations predicts seizure outcome in the individual patient". In: *Scientific Reports* 7.1, p. 13836 (cited on pp. 87, 94).

Fowlkes, Edward B and Colin L Mallows (1983). "A method for comparing two hierarchical clusterings". In: *Journal of the American statistical association* 78.383, pp. 553–569 (cited on p. 112).

Fox, Emily B and Mike West (2011). "Autoregressive models for variance matrices: Stationary inverse Wishart processes". In: *arXiv preprint arXiv:1107.5239* (cited on p. 86).

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani (2001). *The elements of statistical learning*. Vol. 1. Springer series in statistics New York (cited on pp. 1, 90).

– (2008). "Sparse inverse covariance estimation with the graphical lasso". In: *Biostatistics* 9.3, pp. 432–441 (cited on pp. 9, 15–17, 19, 46, 57).

– (2010). "A note on the group lasso and a sparse group lasso". In: *arXiv preprint arXiv:1001.0736* (cited on p. 70).

Friedman, Nir, Michal Linial, Iftach Nachman, and Dana Pe'er (2000). "Using Bayesian networks to analyze expression data". In: *Journal of computational biology* 7.3-4, pp. 601–620 (cited on p. 6).

Geer, Sara van de and Peter Bühlmann (Apr. 2013). "ℓ_0-penalized maximum likelihood for sparse directed acyclic graphs". In: *Ann. Statist.* 41.2, pp. 536–567. DOI: 10.1214/13-AOS1085. URL: https://doi.org/10.1214/13-AOS1085 (cited on p. 10).

Geman, Stuart and Donald Geman (1984). "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images". In: *IEEE Transactions on pattern analysis and machine intelligence* 6, pp. 721–741 (cited on p. 97).

Gibberd, Alex J and Sandipan Roy (2017). "Multiple Changepoint Estimation in High-Dimensional Gaussian Graphical Models". In: *arXiv preprint arXiv:1712.05786* (cited on pp. 19, 20).

Giraud, Christophe (2014). *Introduction to high-dimensional statistics*. Vol. 138. CRC Press (cited on p. 9).

Glanville, Jacob et al. (2009). "Precise determination of the diversity of a combinatorial antibody library gives insight into the human immunoglobulin repertoire". In: *Proceedings of the National Academy of Sciences* 106.48, pp. 20216–20221 (cited on p. 108).

Goldstein, Tom, Christoph Studer, and Richard Baraniuk (2014). "A field guide to forward-backward splitting with a FASTA implementation". In: *arXiv preprint arXiv:1411.3406* (cited on pp. 26, 39).

Gönen, Mehmet and Ethem Alpaydın (2011). "Multiple kernel learning algorithms". In: *Journal of machine learning research* 12.Jul, pp. 2211–2268 (cited on p. 90).

Guo, Jian, Elizaveta Levina, George Michailidis, and Ji Zhu (2011). "Joint estimation of multiple graphical models". In: *Biometrika* 98.1, pp. 1–15 (cited on p. 10).

Guo, Xiaodong, Ting Sun, Mei Yang, Zhiyan Li, Zhiwei Li, and Yuejuan Gao (2013). "Prognostic value of combined aquaporin 3 and aquaporin 5 overexpression in hepatocellular carcinoma". In: *BioMed research international* 2013 (cited on p. 73).

Gupta, Namita T et al. (2015). "Change-O: a toolkit for analyzing large-scale B cell immunoglobulin repertoire sequencing data". In: *Bioinformatics* 31.20, pp. 3356–3358 (cited on pp. 109, 110).

Hallac, David, Jure Leskovec, and Stephen Boyd (2015). "Network Lasso: Clustering and Optimization in Large Graphs". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '15. Sydney, NSW, Australia: ACM, pp. 387–396. ISBN: 978-1-4503-3664-2. DOI: 10.1145/2783258.2783313. URL: http://doi.acm.org/10.1145/2783258.2783313 (cited on p. 20).

Hallac, David, Youngsuk Park, Stephen Boyd, and Jure Leskovec (2017). "Network Inference via the Time-Varying Graphical Lasso". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '17. Halifax, NS, Canada: ACM, pp. 205–213. ISBN: 978-1-4503-4887-4. DOI: 10.1145/3097983.3098037. URL: http://doi.acm.org/10.1145/3097983.3098037 (cited on pp. 10, 19, 20, 36, 37, 39, 40, 43, 46, 47, 51, 53, 57).

Hamming, Richard W (1950). "Error detecting and error correcting codes". In: *Bell Labs Technical Journal* 29.2, pp. 147–160 (cited on pp. 67, 111).

Harville, David A (1997). *Matrix algebra from a statistician's perspective*. Vol. 1. Springer (cited on p. 106).

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The elements of statistical learning*. Vol. 2. 1. Springer (cited on p. 8).

Hastie, Trevor, Robert Tibshirani, and Martin Wainwright (2015). *Statistical learning with sparsity: the lasso and generalizations*. CRC Press (cited on pp. 8, 16).

Hastings, W Keith (1970). "Monte Carlo sampling methods using Markov chains and their applications". In: (cited on p. 97).

Hecker, Michael, Sandro Lambeck, Susanne Toepfer, Eugene Van Someren, and Reinhard Guthke (2009). "Gene regulatory network inference: Data integration in dynamic models: A review". In: *Biosystems* 96.1, pp. 86–103. DOI: https://doi.org/10.1016/j.biosystems.2008.12.004. URL: http://www.sciencedirect.com/science/article/pii/S0303264708002608 (cited on pp. 6, 7, 31, 45, 56).

Hernández-Lobato, Jose Miguel, Daniel Hernández-Lobato, and Alberto Suárez (2011). "Network-based sparse Bayesian classification". In: *Pattern Recognition* 44.4, pp. 886–900 (cited on p. 2).

Heyde, Silvia Von der, Christian Bender, Frauke Henjes, Johanna Sonntag, Ulrike Korf, and Tim Beissbarth (2014). "Boolean ErbB network reconstructions

and perturbation simulations reveal individual drug response in different breast cancer cell lines". In: *BMC systems biology* 8.1, p. 75 (cited on p. 46).

Höller, Yvonne, Raoul Kutil, Lukas Klaffenböck, Aljoscha Thomschewski, Peter M Höller, Arne C Bathke, Julia Jacobs, Alexandra C Taylor, Raffaele Nardone, and Eugen Trinka (2015). "High-frequency oscillations in epilepsy and surgical outcome. A meta-analysis". In: *Frontiers in human neuroscience* 9, p. 574 (cited on p. 87).

Hollmén, Maija, Ping Liu, Kari Kurppa, Hans Wildiers, Irene Reinvall, Thijs Vandorpe, Ann Smeets, Karen Deraedt, Tero Vahlberg, Heikki Joensuu, et al. (2012). "Proteolytic processing of ErbB4 in breast cancer". In: *PLoS One* 7.6, e39413 (cited on p. 71).

Honorio, Jean and Dimitris Samaras (2010). "Multi-Task Learning of Gaussian Graphical Models." In: *ICML*. Citeseer, pp. 447–454 (cited on p. 10).

Horn, Roger A and Charles R Johnson (2012). *Matrix analysis*. CUP (cited on p. 21).

Huang, Lei, Li Liao, and Cathy H Wu (2016). "Inference of protein-protein interaction networks from multiple heterogeneous data". In: *EURASIP Journal on Bioinformatics and Systems Biology* 2016.1, pp. 1–9 (cited on pp. 6, 45).

Hubert, Lawrence and Phipps Arabie (1985). "Comparing partitions". In: *Journal of classification* 2.1, pp. 193–218 (cited on p. 112).

Huszar, Dennis, Maria-Elena Theoclitou, Jeffrey Skolnik, and Ronald Herbst (2009). "Kinesin motor proteins as targets for cancer therapy". In: *Cancer and Metastasis Reviews* 28.1-2, pp. 197–208 (cited on p. 67).

Isensee, Jörg, Henning Witt, Reinhard Pregla, Roland Hetzer, Vera Regitz-Zagrosek, and Patricia Ruiz Noppinger (2008). "Sexually dimorphic gene expression in the heart of mice and men". In: *Journal of molecular medicine* 86.1, pp. 61–74 (cited on p. 82).

Jacob, Laurent, Guillaume Obozinski, and Jean-Philippe Vert (2009). "Group lasso with overlap and graph lasso". In: *Proceedings of the 26th annual international conference on machine learning*. ACM, pp. 433–440 (cited on p. 69).

Jacobs, J, R Staba, E Asano, H Otsubo, JY Wu, M Zijlmans, I Mohamed, P Kahane, F Dubeau, V Navarro, et al. (2012). "High-frequency oscillations (HFOs) in clinical epilepsy". In: *Progress in neurobiology* 98.3, pp. 302–315 (cited on p. 87).

Jalali, A. and S. Sanghavi (2011). "Learning the dependence graph of time series with latent factors". In: *arXiv preprint arXiv:1106.1887* (cited on p. 46).

Janeway, Charles A, Paul Travers, Mark Walport, and Mark J Shlomchik (1997). *Immunobiology: the immune system in health and disease*. Vol. 1. Current Biology Singapore (cited on p. 111).

Jansen, Ronald, Haiyuan Yu, Dov Greenbaum, Yuval Kluger, Nevan J Krogan, Sambath Chung, Andrew Emili, Michael Snyder, Jack F Greenblatt, and Mark Gerstein (2003). "A Bayesian networks approach for predicting protein-protein interactions from genomic data". In: *science* 302.5644, pp. 449–453 (cited on p. 6).

Jiruska, Premysl, Marco de Curtis, John GR Jefferys, Catherine A Schevon, Steven J Schiff, and Kaspar Schindler (2013). "Synchronization and desynchronization in epilepsy: controversies and hypotheses". In: *The Journal of physiology* 591.4, pp. 787–797 (cited on p. 86).

Jones, Beatrix, Carlos Carvalho, Adrian Dobra, Chris Hans, Chris Carter, and Mike West (2005). "Experiments in stochastic computation for high-dimensional graphical models". In: *Statistical Science*, pp. 388–400 (cited on p. 11).

Jones, Donald R (2001). "A taxonomy of global optimization methods based on response surfaces". In: *Journal of global optimization* 21.4, pp. 345–383 (cited on p. 29).

Jozefczuk, Szymon, Sebastian Klie, Gareth Catchpole, Jedrzej Szymanski, Alvaro Cuadros-Inostroza, Dirk Steinhauser, Joachim Selbig, and Lothar Willmitzer (2010). "Metabolomic and transcriptomic stress response of Escherichia coli". In: *Molecular systems biology* 6.1, p. 364 (cited on pp. 59, 60).

Kaestner, Phillip and Holger Bastians (2010). "Mitotic drug targets". In: *Journal of cellular biochemistry* 111.2, pp. 258–265 (cited on p. 68).

Kanehisa, Minoru (2001). "Prediction of higher order functional networks from genomic data". In: *Pharmacogenomics* 2.4, pp. 373–385 (cited on p. 6).

Khongkow, P, AR Gomes, C Gong, EPS Man, J WH Tsang, F Zhao, LJ Monteiro, RC Coombes, RH Medema, US Khoo, et al. (2016). "Paclitaxel targets FOXM1 to regulate KIF20A in mitotic catastrophe and breast cancer paclitaxel resistance". In: *Oncogene* 35.8, pp. 990–1002 (cited on p. 68).

Kim, Ji-Yeon, Hae Hyun Jung, In-Gu Do, SooYoun Bae, Se Kyung Lee, Seok Won Kim, Jeong Eon Lee, Seok Jin Nam, Jin Seok Ahn, Yeon Hee Park, et al. (2016). "Prognostic value of ERBB4 expression in patients with triple negative breast cancer". In: *BMC cancer* 16.1, p. 138 (cited on p. 71).

Kleinstein, Steven H, Yoram Louzoun, and Mark J Shlomchik (2003). "Estimating hypermutation rates from clonal tree data". In: *The Journal of Immunology* 171.9, pp. 4639–4649 (cited on p. 109).

Kolar, Mladen, Le Song, Amr Ahmed, Eric P Xing, et al. (2010). "Estimating time-varying networks". In: *The Annals of Applied Statistics* 4.1, pp. 94–123 (cited on p. 10).

Lachmann, Alexander, Federico M Giorgi, Gonzalo Lopez, and Andrea Califano (2016). "ARACNe-AP: gene network reverse engineering through adaptive partitioning inference of mutual information". In: *Bioinformatics* 32.14, pp. 2233–2235 (cited on p. 7).

Lanckriet, Gert RG, Tijl De Bie, Nello Cristianini, Michael I Jordan, and William Stafford Noble (2004). "A statistical framework for genomic data fusion". In: *Bioinformatics* 20.16, pp. 2626–2635 (cited on p. 90).

Langfelder, Peter and Steve Horvath (2008). "WGCNA: an R package for weighted correlation network analysis". In: *BMC bioinformatics* 9.1, p. 559 (cited on p. 7).

Lauritzen, Steffen L (1996). *Graphical models*. Vol. 17. Clarendon Press (cited on pp. 12, 14, 15, 46, 106).

Leandro-Garcia, Luis J, Susanna Leskelä, Carlos Jara, Henrik Gréen, Elisabeth Åvall-Lundqvist, Heather E Wheeler, M Eileen Dolan, Lucia Inglada-Perez, Agnieszka Maliszewska, Aguirre A de Cubas, et al. (2012). "Regulatory polymorphisms in β-tubulin IIa are associated with paclitaxel-induced peripheral neuropathy". In: *Clinical Cancer Research* 18.16, pp. 4441–4448 (cited on p. 68).

Lenkoski, A and A Dobra (2008). *Bayesian structural learning and estimation in Gaussian graphical models* (cited on p. 11).

Li, Ang, Dehong Lu, Yupeng Zhang, Jia Li, Yu Fang, Fei Li, and Jiabang Sun (2013). "Critical role of aquaporin-3 in epidermal growth factor-induced migration of colorectal carcinoma cells and its clinical significance". In: *Oncology reports* 29.2, pp. 535–540 (cited on p. 73).

Lis, Raphael, Charles C Karrasch, Michael G Poulos, Balvir Kunar, David Redmond, Jose G Barcia Duran, Chaitanya R Badwe, William Schachterle, Michael Ginsberg, Jenny Xiang, et al. (2017). "Conversion of adult endothelium to immunocompetent haematopoietic stem cells". In: *Nature* 545.7655, p. 439 (cited on pp. 80, 81).

Liu, Han, Fang Han, and Cun-hui Zhang (2012). "Transelliptical graphical models". In: *Advances in neural information processing systems*, pp. 800–808 (cited on p. 45).

Liu, Han, John Lafferty, and Larry Wasserman (2009). "The nonparanormal: Semiparametric estimation of high dimensional undirected graphs". In: *Journal of Machine Learning Research* 10.Oct, pp. 2295–2328 (cited on p. 22).

Liu, Han, Fang Han, Ming Yuan, John Lafferty, Larry Wasserman, et al. (2012). "High-dimensional semiparametric Gaussian copula graphical models". In: *The Annals of Statistics* 40.4, pp. 2293–2326 (cited on p. 22).

Lozano, Aurélie C, Naoki Abe, Yan Liu, and Saharon Rosset (2009). "Grouped graphical Granger modeling for gene expression regulatory networks discovery". In: *Bioinformatics* 25.12, pp. i110–i118 (cited on pp. 6, 45).

Ma, Shiqian, Lingzhou Xue, and Hui Zou (2013). "Alternating Direction Methods for Latent Variable Gaussian Graphical Model Selection". In: *Neural Computation* 25.8, pp. 2172–2198. ISSN: 0899-7667. DOI: 10.1162/NECO_a_00379 (cited on pp. 21, 47, 53, 57).

Mallat, Stéphane (1999). *A wavelet tour of signal processing*. Elsevier (cited on p. 89).

Margolin, Adam A, Ilya Nemenman, Katia Basso, Chris Wiggins, Gustavo Stolovitzky, Riccardo Dalla Favera, and Andrea Califano (2006). "ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context". In: *BMC bioinformatics* 7.1, S7 (cited on pp. 7, 65, 67).

Marlar, Saw, Helene H Jensen, Frédéric H Login, and Lene N Nejsum (2017). "Aquaporin-3 in cancer". In: *International journal of molecular sciences* 18.10, p. 2106 (cited on p. 73).

Mascelli, Samantha, Annalisa Barla, Alessandro Raso, Sofia Mosci, Paolo Nozza, Roberto Biassoni, Giovanni Morana, Martin Huber, Cristian Mircean, Daniel Fasulo, et al. (2013). "Molecular fingerprinting reflects different histotypes

and brain region in low grade gliomas". In: *BMC cancer* 13.1, p. 387 (cited on p. 9).

Meinshausen, Nicolai and Peter Bühlmann (2006). "High-dimensional graphs and variable selection with the lasso". In: *The annals of statistics*, pp. 1436–1462 (cited on pp. 9, 15–17).

Meng, Zhaoshi, Brian Eriksson, and Al Hero (2014). "Learning latent variable Gaussian graphical models". In: *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 1269–1277 (cited on p. 46).

Mercier, Manuel R, Stephan Bickel, Pierre Megevand, David M Groppe, Charles E Schroeder, Ashesh D Mehta, and Fred A Lado (2017). "Evaluation of cortical local field potential diffusion in stereotactic electro-encephalography recordings: A glimpse on white matter signal". In: *Neuroimage* 147, pp. 219–232 (cited on p. 92).

Meyer, Patrick E, Frederic Lafitte, and Gianluca Bontempi (2008). "minet: A Bioconductor package for inferring large transcriptional networks using mutual information". In: *BMC bioinformatics* 9.1, p. 461 (cited on p. 7).

Moghaddam, Baback, Emtiyaz Khan, Kevin P Murphy, and Benjamin M Marlin (2009). "Accelerating Bayesian structural inference for non-decomposable Gaussian graphical models". In: *Advances in Neural Information Processing Systems*, pp. 1285–1293 (cited on pp. 10, 11).

Mohan, Karthik, Mike Chung, Seungyeop Han, Daniela Witten, Su-In Lee, and Maryam Fazel (2012). "Structured learning of Gaussian graphical models". In: *Advances in neural information processing systems*, pp. 620–628 (cited on p. 20).

Molinaro, Annette M, Richard Simon, and Ruth M Pfeiffer (2005). "Prediction error estimation: a comparison of resampling methods". In: *Bioinformatics* 21.15, pp. 3301–3307 (cited on pp. 23, 27).

Molinelli, Evan J, Anil Korkut, Weiqing Wang, Martin L Miller, Nicholas P Gauthier, Xiaohong Jing, Poorvi Kaushik, Qin He, Gordon Mills, David B Solit, et al. (2013). "Perturbation biology: inferring signaling networks in cellular systems". In: *PLoS computational biology* 9.12 (cited on p. 46).

Mooij, Anne H, Birgit Frauscher, Mina Amiri, Willem M Otte, and Jean Gotman (2016). "Differentiating epileptic from non-epileptic high frequency intracerebral EEG signals with measures of wavelet entropy". In: *Clinical Neurophysiology* 127.12, pp. 3529–3536 (cited on p. 87).

Mosci, Sofia, Annalisa Barla, Alessandro Verri, and Lorenzo Rosasco (2008). "Finding structured gene signatures". In: *Bioinformatics and Biomedicine Workshops, 2008. BIBMW 2008. IEEE International Conference on*. IEEE, pp. 158–165 (cited on p. 9).

Mukhopadhyay, Partha, Subhankar Chakraborty, Moorthy P Ponnusamy, Imayavaramban Lakshmanan, Maneesh Jain, and Surinder K Batra (2011). "Mucins in the pathogenesis of breast cancer: implications in diagnosis, prognosis and therapy". In: *Biochimica et Biophysica Acta (BBA)-Reviews on Cancer* 1815.2, pp. 224–240 (cited on p. 73).

Murphy, Kevin P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press (cited on pp. 11, 17, 106).

Murray, Iain, Ryan Prescott Adams, and David JC MacKay (2010). "Elliptical slice sampling". In: JMLR: W&CP 9, pp. 541–548 (cited on p. 97).

Omerhodzic, Ibrahim, Samir Avdakovic, Amir Nuhanovic, and Kemal Dizdarevic (2013). "Energy distribution of EEG signals: EEG signal wavelet-neural network classifier". In: *arXiv preprint arXiv:1307.7897* (cited on p. 87).

Oprea, Mihaela L (1999). "Antibody repertoires and pathogen recognition: the role of germline diversity and somatic hypermutation". PhD thesis. Citeseer (cited on p. 109).

Orchard, Peter, Felix Agakov, and Amos Storkey (2013). "Bayesian inference in sparse Gaussian graphical models". In: *arXiv preprint arXiv:1309.7311* (cited on p. 45).

Pournara, Iosifina and Lorenz Wernisch (2007). "Factor analysis for gene regulatory networks and transcription factor activity profiles". In: *BMC bioinformatics* 8.1, p. 61 (cited on p. 10).

Ralph, Duncan K and Frederick A Matsen IV (2016). "Consistency of VDJ rearrangement and substitution parameters enables accurate B cell receptor sequence annotation". In: *PLoS Comput Biol* 12.1, e1004409 (cited on p. 109).

Rao, Arvind, AO Hero, James Douglas Engel, et al. (2007). "Using directed information to build biologically relevant influence networks". In: *Proc. Computational Systems Bioinformatics (CSB)*, pp. 145–156 (cited on p. 7).

Rasmussen, Carl Edward (2004). "Gaussian processes in machine learning". In: *Advanced lectures on machine learning*. Springer, pp. 63–71 (cited on pp. 30, 95).

Ravikumar, Pradeep, Martin J Wainwright, Garvesh Raskutti, and Bin Yu (2011). "High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence". In: *Electronic Journal of Statistics* 5, pp. 935–980 (cited on pp. 9, 15, 16).

Rock, Edwin P et al. (1994). "CDR3 length in antigen-specific immune receptors." In: *The Journal of experimental medicine* 179.1, pp. 323–328 (cited on p. 108).

Rosso, Osvaldo A, Susana Blanco, Juliana Yordanova, Vasil Kolev, Alejandra Figliola, Martin Schürmann, and Erol Başar (2001). "Wavelet entropy: a new tool for analysis of short duration brain electrical signals". In: *Journal of neuroscience methods* 105.1, pp. 65–75 (cited on p. 87).

Rothman, Adam J, Peter J Bickel, Elizaveta Levina, Ji Zhu, et al. (2008). "Sparse permutation invariant covariance estimation". In: *Electronic Journal of Statistics* 2, pp. 494–515 (cited on p. 16).

Roverato, Alberto (2002). "Hyper inverse Wishart distribution for non-decomposable graphs and its application to Bayesian inference for Gaussian graphical models". In: *Scandinavian Journal of Statistics* 29.3, pp. 391–411 (cited on p. 11).

Sabatti, Chiara and Gareth M James (2005). "Bayesian sparse hidden components analysis for transcription regulation networks". In: *Bioinformatics* 22.6, pp. 739–746 (cited on p. 10).

Sahu, Ankita, PK Patra, Manoj Kumar Yadav, and Meena Varma (2017). "Identification and characterization of ErbB4 kinase inhibitors for effective breast cancer therapy". In: *Journal of Receptors and Signal Transduction* 37.5, pp. 470–480 (cited on p. 71).

Salzo, Saverio (2017). "The variable metric forward-backward splitting algorithm under mild differentiability assumptions". In: *SIAM Journal on Optimization* 27.4, pp. 2153–2181 (cited on pp. 25, 26, 35, 36, 38, 39).

Sandberg, Rickard (2014). "Entering the era of single-cell transcriptomics in biology and medicine". In: *Nature methods* 11.1, p. 22 (cited on p. 79).

Sandler, Vladislav M, Raphael Lis, Ying Liu, Alon Kedem, Daylon James, Olivier Elemento, Jason M Butler, Joseph M Scandura, and Shahin Rafii (2014). "Reprogramming human endothelial cells to haematopoietic cells requires vascular induction". In: *Nature* 511.7509, p. 312 (cited on p. 81).

Sanguinetti, Guido, Magnus Rattray, and Neil D Lawrence (2006). "A probabilistic dynamical model for quantitative inference of the regulatory mechanism of transcription". In: *Bioinformatics* 22.14, pp. 1753–1759 (cited on p. 10).

Satooka, Hiroki and Mariko Hara-Chikuma (2016). "Aquaporin-3 controls breast cancer cell migration by regulating hydrogen peroxide transport and its downstream cell signaling". In: *Molecular and cellular biology*, MCB–00971 (cited on p. 73).

Sculley, David (2010). "Web-scale k-means clustering". In: *Proceedings of the 19th international conference on World wide web*. ACM, pp. 1177–1178 (cited on p. 110).

Shannon, Paul, Andrew Markiel, Owen Ozier, Nitin S Baliga, Jonathan T Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker (2003). "Cytoscape: a software environment for integrated models of biomolecular interaction networks". In: *Genome research* 13.11, pp. 2498–2504 (cited on p. 8).

Silver, Matt, Peng Chen, Ruoying Li, Ching-Yu Cheng, Tien-Yin Wong, E-Shyong Tai, Yik-Ying Teo, and Giovanni Montana (2013). "Pathways-driven sparse regression identifies pathways and genes associated with high-density lipoprotein cholesterol in two Asian cohorts". In: *PLoS genetics* 9.11, e1003939 (cited on p. 9).

Sima, Chao, Jianping Hua, and Sungwon Jung (2009). "Inference of gene regulatory networks using time-series data: a survey". In: *Current genomics* 10.6, pp. 416–429 (cited on p. 46).

Skylaki, Stavroula, Oliver Hilsenbeck, and Timm Schroeder (2016). "Challenges in long-term imaging and quantification of single-cell dynamics". In: *Nature biotechnology* 34.11, p. 1137 (cited on p. 80).

Smith, Temple F and Michael S Waterman (1981). "Identification of common molecular subsequences". In: *Journal of molecular biology* 147.1, pp. 195–197 (cited on p. 111).

Snoek, Jasper, Hugo Larochelle, and Ryan P Adams (2012). "Practical Bayesian optimization of machine learning algorithms". In: *Advances in neural information processing systems*, pp. 2951–2959 (cited on pp. 29, 30, 43, 59).

Soriano, Miguel C, Guiomar Niso, Jillian Clements, Silvia Ortín, Sira Carrasco, María Gudín, Claudio R Mirasso, and Ernesto Pereda (2017). "Automated Detection of Epileptic Biomarkers in Resting-State Interictal MEG Data". In: *Frontiers in neuroinformatics* 11, p. 43 (cited on p. 87).

Spitzer, Matthew H and Garry P Nolan (2016). "Mass cytometry: single cells, many features". In: *Cell* 165.4, pp. 780–791 (cited on p. 79).

Staba, Richard J, Matt Stead, and Gregory A Worrell (2014). "Electrophysiological biomarkers of epilepsy". In: *Neurotherapeutics* 11.2, pp. 334–346 (cited on p. 87).

Steuer, Ralf, Jürgen Kurths, Carsten O Daub, Janko Weise, and Joachim Selbig (2002). "The mutual information: detecting and evaluating dependencies between variables". In: *Bioinformatics* 18.suppl 2, S231–S240 (cited on p. 7).

Strehl, Alexander and Joydeep Ghosh (2002). "Cluster ensembles—a knowledge reuse framework for combining multiple partitions". In: *Journal of machine learning research* 3.Dec, pp. 583–617 (cited on p. 112).

Stuart, Joshua M, Eran Segal, Daphne Koller, and Stuart K Kim (2003). "A gene-coexpression network for global discovery of conserved genetic modules". In: *science* 302.5643, pp. 249–255 (cited on p. 7).

Sugimura, Ryohichi, Deepak Kumar Jha, Areum Han, Clara Soria-Valles, Edroaldo Lummertz da Rocha, Yi-Fen Lu, Jeremy A Goettel, Erik Serrao, R Grant Rowe, Mohan Malleshaiah, et al. (2017). "Haematopoietic stem and progenitor cells from human pluripotent stem cells". In: *Nature* 545.7655, p. 432 (cited on p. 82).

Sundvall, Maria, Kristiina Iljin, Sami Kilpinen, Henri Sara, Olli-Pekka Kallioniemi, and Klaus Elenius (2008). "Role of ErbB4 in breast cancer". In: *Journal of mammary gland biology and neoplasia* 13.2, pp. 259–268 (cited on p. 71).

Tang, Careen K, Xiao-Zheng Wu Concepcion, Melissa Milan, Xiaoqi Gong, Elizabeth Montgomery, and Marc E Lippman (1999). "Ribozyme-mediated down-regulation of ErbB-4 in estrogen receptor-positive breast cancer cells inhibits proliferation both in vitro and in vivo". In: *Cancer research* 59.20, pp. 5315–5322 (cited on p. 71).

Tang, Fuchou, Catalin Barbacioru, Yangzhou Wang, Ellen Nordman, Clarence Lee, Nanlan Xu, Xiaohui Wang, John Bodeau, Brian B Tuch, Asim Siddiqui, et al. (2009). "mRNA-Seq whole-transcriptome analysis of a single cell". In: *Nature methods* 6.5, p. 377 (cited on p. 79).

Tibshirani, Robert (1996). "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288 (cited on pp. 16, 92).

Tomasi, Federico, Veronica Tozzo, Alessandro Verri, and Saverio Salzo (2018a). "Forward-Backward Splitting for Time-Varying Graphical Models". In: *Proceedings of the Ninth International Conference on Probabilistic Graphical Models*. Ed. by Václav Kratochvíl and Milan Studený. Vol. 72. Proceedings of Machine Learning Research. Prague, Czech Republic: PMLR, pp. 475–486.

URL: http://proceedings.mlr.press/v72/tomasi18a.html (cited on p. 34).

Trapnell, Cole (2015). "Defining cell types and states with single-cell genomics". In: *Genome research* 25.10, pp. 1491–1498 (cited on p. 80).

Vander Heiden, Jason A et al. (2014). "pRESTO: a toolkit for processing high-throughput sequencing raw reads of lymphocyte receptor repertoires". In: *Bioinformatics*, btu138 (cited on p. 109).

Varoquaux, Gaël, Alexandre Gramfort, Jean-Baptiste Poline, and Bertrand Thirion (2010). "Brain covariance selection: better individual functional connectivity models using population prior". In: *Advances in neural information processing systems*, pp. 2334–2342 (cited on p. 10).

Vila-Vidal, Manel, Alessandro Principe, Miguel Ley, Gustavo Deco, Adrià Tauste Campo, and Rodrigo Rocamora (2017). "Detection of recurrent activation patterns across focal seizures: Application to seizure onset zone identification". In: *Clinical Neurophysiology* 128.6, pp. 977–985 (cited on p. 87).

Villa, Silvia, Lorenzo Rosasco, Sofia Mosci, and Alessandro Verri (2014). "Proximal methods for the latent group lasso penalty". In: *Computational Optimization and Applications* 58.2, pp. 381–407 (cited on p. 69).

Vinh, Nguyen Xuan, Julien Epps, and James Bailey (2009). "Information theoretic measures for clusterings comparison: is a correction for chance necessary?" In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, pp. 1073–1080 (cited on p. 112).

Vojta, Aleksandar, Ivana Samaržija, Luka Bočkor, and Vlatka Zoldoš (2016). "Glyco-genes change expression in cancer through aberrant methylation". In: *Biochimica et Biophysica Acta (BBA)-General Subjects* 1860.8, pp. 1776–1785 (cited on p. 73).

Wahlster, Lara and George Q Daley (2016). "Progress towards generation of human haematopoietic stem cells". In: *Nature cell biology* 18.11, p. 1111 (cited on p. 80).

Wainwright, Martin J, Pradeep Ravikumar, and John D Lafferty (2007). "High-Dimensional Graphical Model Selection Using ℓ_1-Regularized Logistic Regression". In: *Advances in neural information processing systems* 19, p. 1465 (cited on pp. 9, 15, 17).

Wang, Shuhui, Shuqiang Jiang, Qingming Huang, and Qi Tian (2010). "S3MKL: scalable semi-supervised multiple kernel learning for image data mining". In: *Proceedings of the 18th ACM international conference on Multimedia*. ACM, pp. 163–172 (cited on p. 89).

Weinberger, Kilian Q, Fei Sha, Qihui Zhu, and Lawrence K Saul (2007). "Graph Laplacian regularization for large-scale semidefinite programming". In: *Advances in neural information processing systems*, pp. 1489–1496 (cited on p. 20).

Wen, Fei, Yuan Yang, Peilin Liu, and Robert C Qiu (2016). "Positive Definite Estimation of Large Covariance Matrix Using Generalized Nonconvex Penalties". In: *IEEE Access* 4, pp. 4168–4182 (cited on p. 10).

Wilson, Andrew Gordon and Zoubin Ghahramani (2011). "Generalised Wishart processes". In: *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*. AUAI Press, pp. 736–744 (cited on pp. 86, 96, 97, 103).

Witten, Daniela M. and Robert Tibshirani (2009). "Covariance-regularized regression and classification for high dimensional problems". In: *Journal of the Royal Statistical Society. Series B: Statistical Methodology* 71.3, pp. 615–636. ISSN: 13697412. DOI: 10.1111/j.1467-9868.2009.00699.x. URL: http://doi.wiley.com/10.1111/j.1467-9868.2009.00699.x (cited on p. 51).

Xie, Yuying, Yufeng Liu, and William Valdar (2016). "Joint estimation of multiple dependent Gaussian graphical models with applications to mouse genomics". In: *Biometrika* 103.3, pp. 493–511 (cited on p. 10).

Yaari, Gur et al. (2013). "Models of Somatic Hypermutation Targeting and Substitution Based on Synonymous Mutations from High-Throughput Immunoglobulin Sequencing Data". In: *Frontiers in Immunology* 4 (cited on p. 111).

Yardi, Ruta, Juan Bulacio, William Bingaman, Jorge Gonzalez-Martinez, Imad Najm, and Lara Jehi (2016). "Interictal Spikes on Intracranial EEG as a Potential Biomarker of Epilepsy Severity (P4. 071)". In: *Neurology* 86.16 Supplement, P4–071 (cited on p. 87).

Yuan, Guo-Cheng, Long Cai, Michael Elowitz, Tariq Enver, Guoping Fan, Guoji Guo, Rafael Irizarry, Peter Kharchenko, Junhyong Kim, Stuart Orkin, et al. (2017). "Challenges and emerging directions in single-cell analysis". In: *Genome biology* 18.1, p. 84 (cited on p. 80).

Yuan, Ming (2012). "Discussion: Latent variable graphical model selection via convex optimization". In: *Ann. Statist.* 40.4, pp. 1968–1972. DOI: 10.1214/12-AOS979. URL: http://dx.doi.org/10.1214/12-AOS979 (cited on pp. 46, 55).

Yuan, Ming and Yi Lin (2006). "Model selection and estimation in regression with grouped variables". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68.1, pp. 49–67 (cited on p. 69).

– (2007). "Model selection and estimation in the Gaussian graphical model". In: *Biometrika*, pp. 19–35 (cited on pp. 9, 15, 17).

Zenobi, Renato (2013). "Single-cell metabolomics: analytical and biological perspectives". In: *Science* 342.6163, p. 1243259 (cited on p. 79).

Zhu, Yun, Lacey L Sullivan, Sujit S Nair, Christopher C Williams, Arvind K Pandey, Luis Marrero, Ratna K Vadlamudi, and Frank E Jones (2006). "Coregulation of estrogen receptor by ERBB4/HER4 establishes a growth-promoting autocrine signal in breast tumor cells". In: *Cancer research* 66.16, pp. 7991–7998 (cited on p. 71).

Zhu, Zhengcai, Lianghe Jiao, Tao Li, Honggang Wang, Wei Wei, and Haixin Qian (2018). "Expression of AQP3 and AQP5 as a prognostic marker in triple-negative breast cancer". In: *Oncology letters* 16.2, pp. 2661–2667 (cited on p. 73).

Zijlmans, Maeike, Premysl Jiruska, Rina Zelmann, Frans SS Leijten, John GR Jefferys, and Jean Gotman (2012). "High-frequency oscillations as a new biomarker in epilepsy". In: *Annals of neurology* 71.2, pp. 169–178 (cited on p. 87).

Zoppoli, Pietro, Sandro Morganella, and Michele Ceccarelli (2010). "TimeDelay-ARACNE: Reverse engineering of gene networks from time-course data by an information theoretic approach". In: *BMC bioinformatics* 11.1, p. 154 (cited on p. 8).

Zou, Hui and Trevor Hastie (2005). "Regularization and variable selection via the elastic net". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.2, pp. 301–320 (cited on p. 91).

Zycinski, Grzegorz, Annalisa Barla, Margherita Squillario, Tiziana Sanavia, Barbara Di Camillo, and Alessandro Verri (2013). "Knowledge Driven Variable Selection (KDVS)–a new approach to enrichment analysis of gene signatures obtained from high–throughput data". In: *Source Code for Biology and Medicine* 8.1, p. 2 (cited on p. 64).

List of Abbreviations

ADMM alternating direction methods of multipliers

BA balanced accuracy

BRCA breast invasive carcinoma

cdf cumulative distribution function

CWT continuous wavelet transform

EI expected improvement

EZ epileptic zone

FBS forward-backward splitting

FNR false negative rate

FPR false positive rate

GDA Gaussian discriminant analysis

GGM Gaussian graphical model

GP Gaussian process

GWP generalised Wishart process

HFO high-frequency oscillation

HSC haematopoietic stem cell

i.i.d. independent and identically distributed

KFCV k-fold cross-validation

LTGL latent variable time-varying graphical lasso

MAP maximum a posteriori

MCCV Monte Carlo cross-validation

MKL multiple kernel learning

MLE maximum likelihood estimation

MSE mean squared error

MT-MKL multi-task multiple kernel learning

MVN multivariate normal

pdf probability density function

PLV phase locking value

PPI protein-protein interaction

QDA quadratic discriminant analysis

SEEG stereo-electroencephalography

TGL time-varying graphical lasso

TNR true negative rate

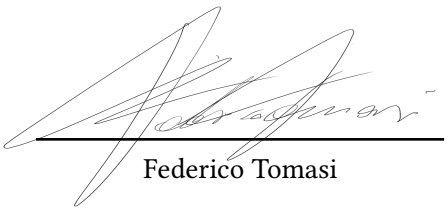
TPR true positive rate

WSD weighted spectral distribution

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Genova, Italy
March 2019



Federico Tomasi